Medical image enhancement

For whom, why, and how to.

For whom?

- Human processing
- Computer processing

Why?

- Can't distinguish between tissues
- Data too noisy for computer algorithm to perform well
- Acquisition or reconstruction artifacts interfere with visualization or algorithm processing

2D/3D image processing tools

Many general purpose tools (or filters) exist to remove noise, increase contrast, etc.

Most IP applications must employ a suite of such tools to achieve their goal.

Linear, shift-invariant filters

- Obey principle of superposition

  $F(x) + F(y) = F(x+y)$
- Are not position-dependent
- Can be implemented in either spatial domain (by convolution) or frequency domain (by weighting components)

How to filter to …

- Increase contrast
- Remove noise
- Emphasize edges
- Detect edges
- Modify shapes

Increase contrast

- Scaling (shift-invariant, linear)
- Window/level
- Histogram equalization (shift-invariant, but nonlinear)
- Adaptive histogram equalization

Increase contrast - scaling

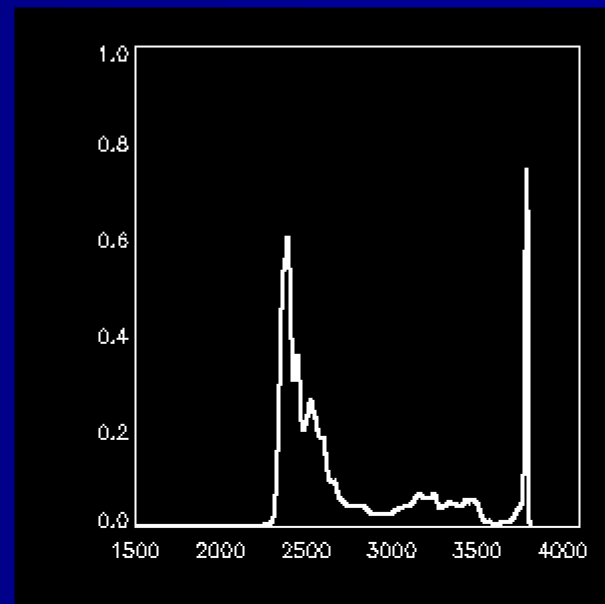
Scaling simply spreads or contracts image intensity values into a new range – e.g. to monitor display range.

Scaling, if the final output is required to have integer values, may not be a one-to-one operation.

Before scaling, consider cropping the image to the region of interest. This can also help your visual system.

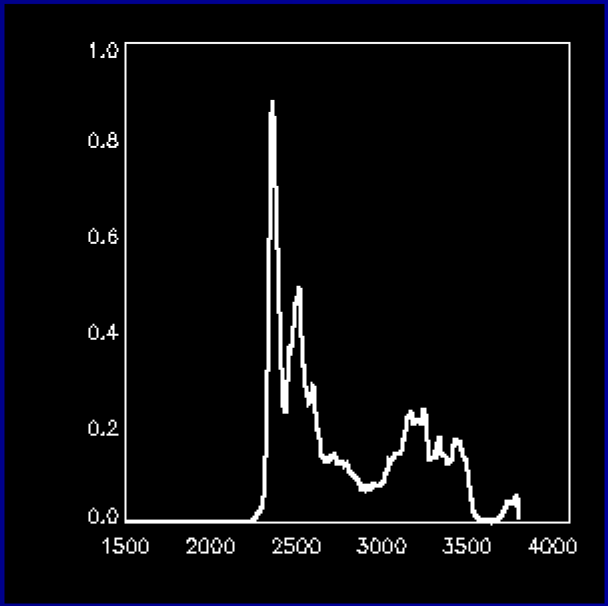
Increase contrast – linear rescaling

Digitized film, 12 bit storage, scaled to 256 colors



Increase contrast – linear rescaling

Simply-cropped image, 12 bit storage, scaled to 256 colors



Increase contrast – window/level

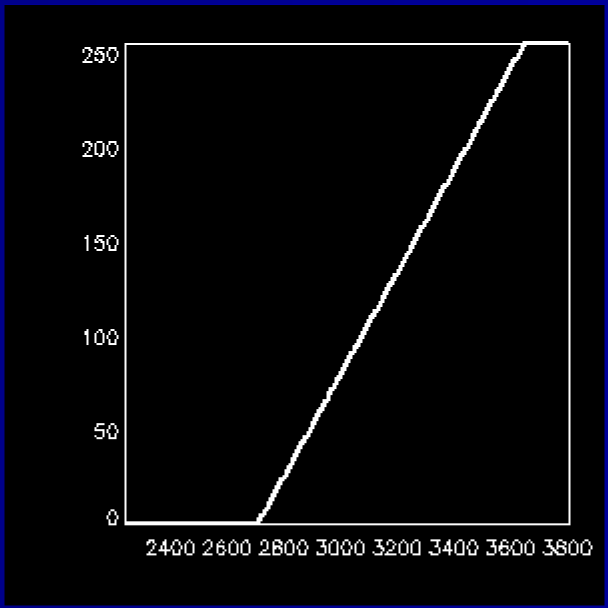
Window and level adjustments are piecewise-linear or nonlinear operations that can be approximated by adjusting the lookup table of a displayed image.

Except on specialized workstations, this will not recover contrast that is lost when the data range of an image was reduced for display.

More properly, contrast adjustment is performed on original data using a lookup table and then mapped to a display.

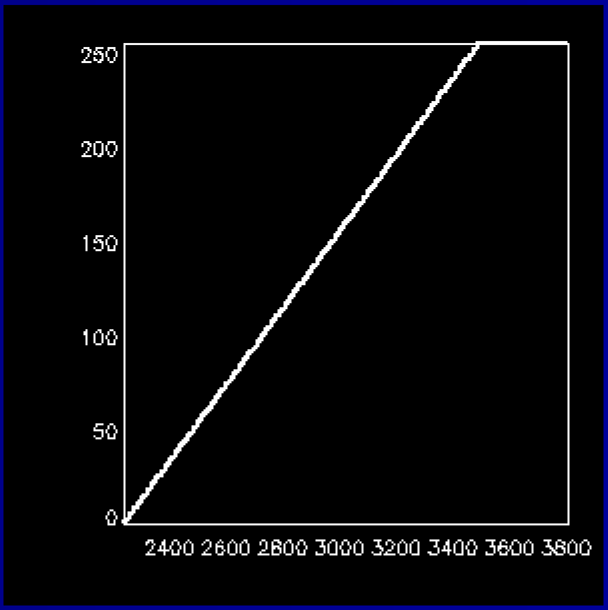
Increase contrast – window/level

Window is narrowed to span less of input data; level (midpoint of slope) is shifted right to decrease brightness.



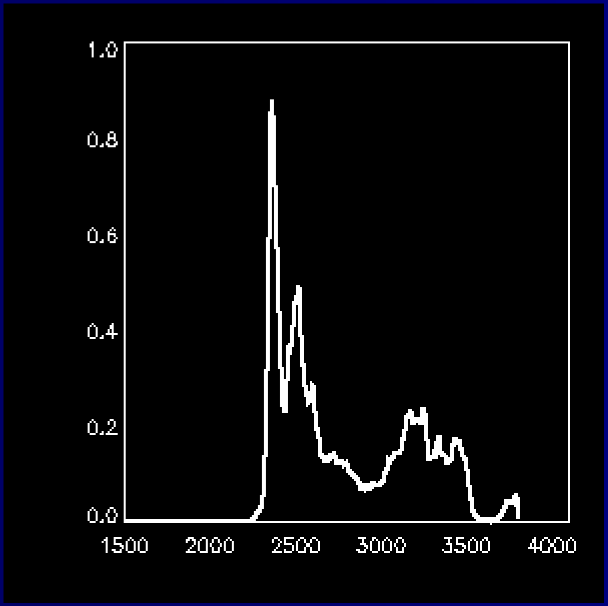
Increase contrast – window/level

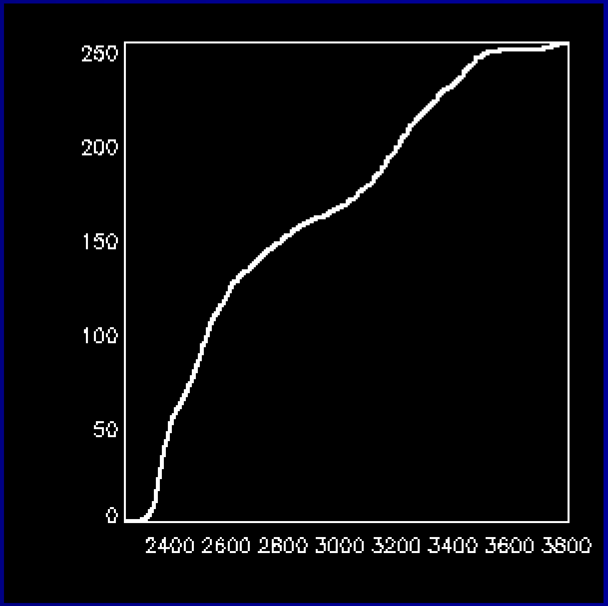
Window is narrowed to span less of input data; level (midpoint of slope) is shifted left to increase brightness.



Increase contrast – histograms

Histogram equalization attempts to take advantage of unassigned output values. It is a one-to-one operation, unlike window/level operations. The lookup table simply follows the shape of the cumulative histogram.

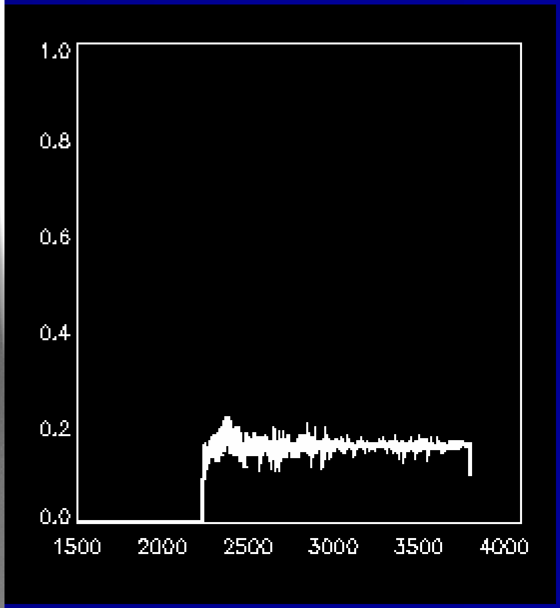




Increase contrast – histogram equalization

Raw image

Global HEQ



Increase contrast – adaptive histogram equalization

Adaptive histogram equalization computes an output intensity level from the histogram of a local neighborhood of each pixel. Many variants exist that attempt to control contrast, interpolate for speed, etc.

Global HEQ

Fully adaptive HEQ

Increase contrast – adaptive histogram equalization

Contrast limitation in histogram equalization means setting a maximum spread of output intensities for adjacent input intensities.

Raw image

Contrast-limited fully adaptive HEQ

How to filter to …

- Increase contrast
- Remove noise
- Emphasize edges
- Detect edges
- Modify shapes

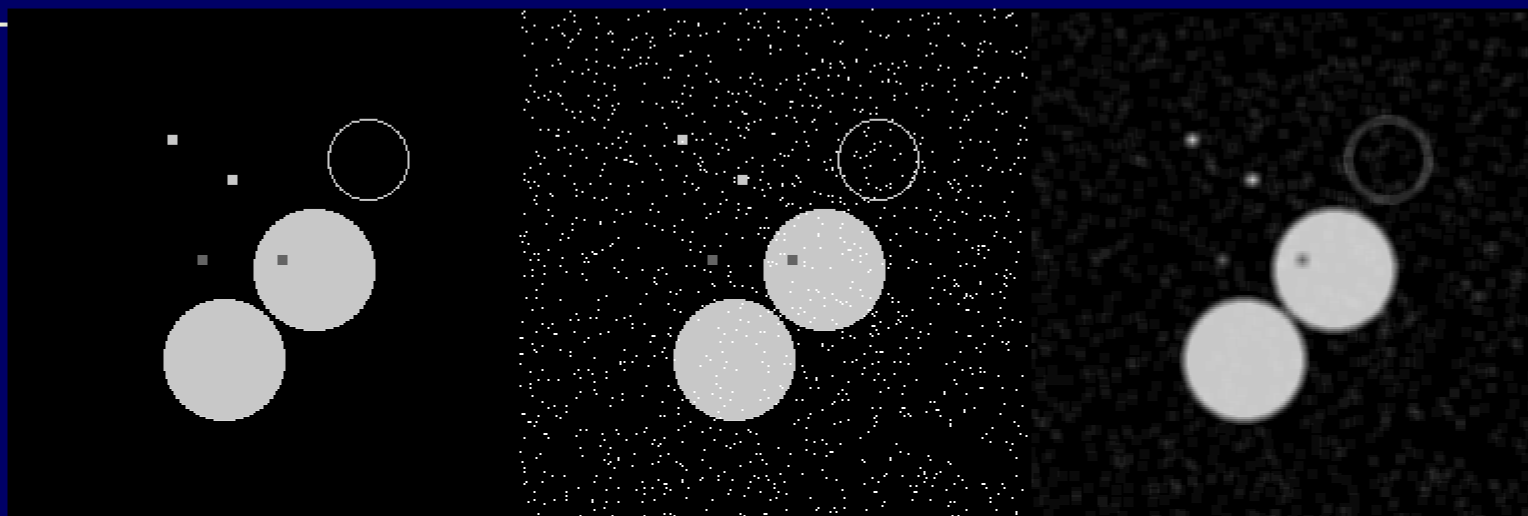
Noise removal – mean filters

Simple convolution filters (linear)

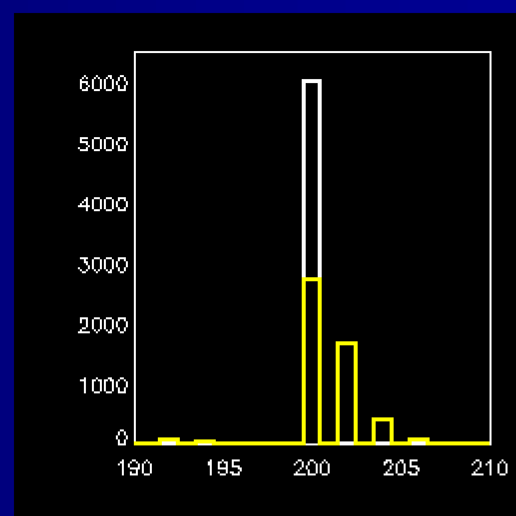
- Replace center pixel with weighted combination of local neighbors; a moving average
- Mean or "boxcar" filter; uniform weights
- Gaussian kernel; center weights dominate

Noise removal – mean filters

Test object With additive noise Boxcar mean, 5x5

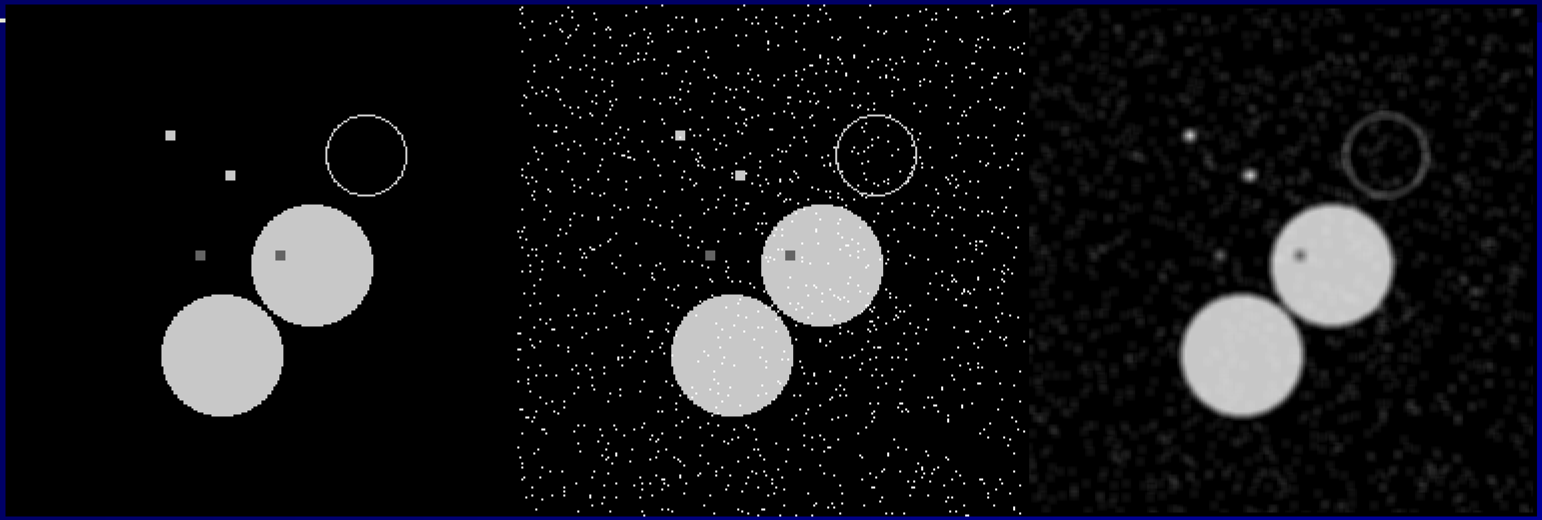


Plot shows histograms of test object and filtered image. Additive noise is brighter than all objects.

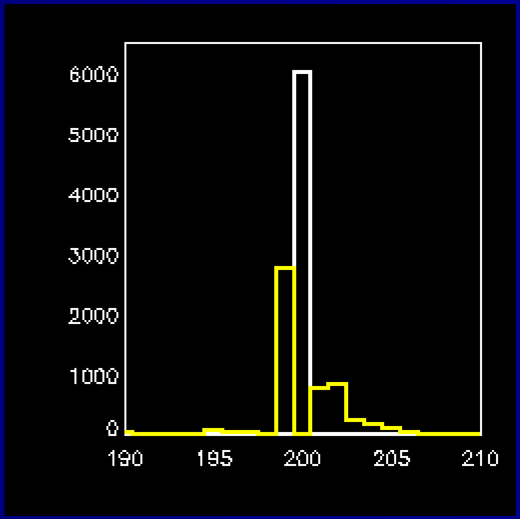


Noise removal – mean filters

Test object | With additive noise | Gaussian mean, fwhm 5



Distribution of intensity values after smoothing differs from simple mean.



Noise removal – mean filters

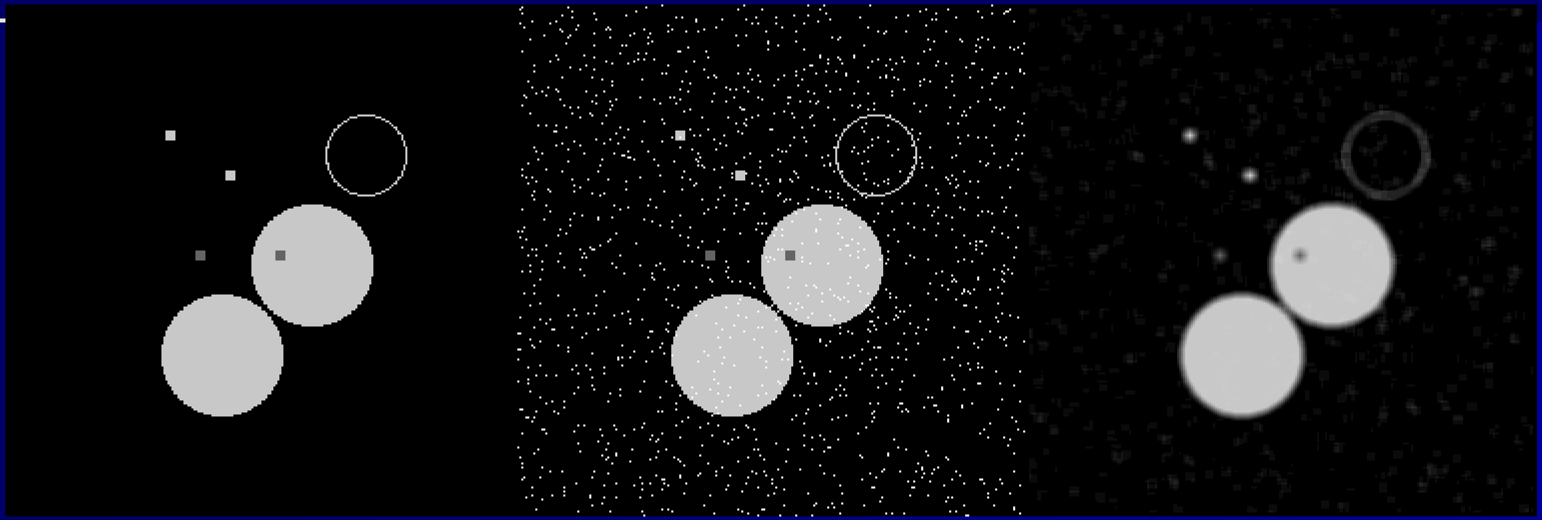
Trimmed mean filter (nonlinear)

- Boxcar kernel
- Discard highest and lowest values
- Compute mean of remaining values

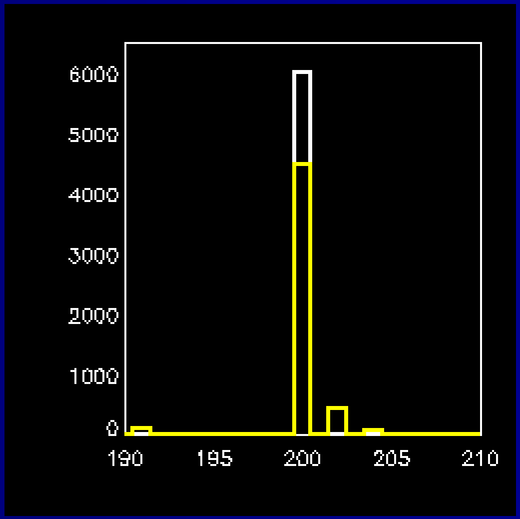
Sometimes called an “Olympic” filter

Noise removal – mean filters

Test object With additive noise Trimmed mean, 5x5

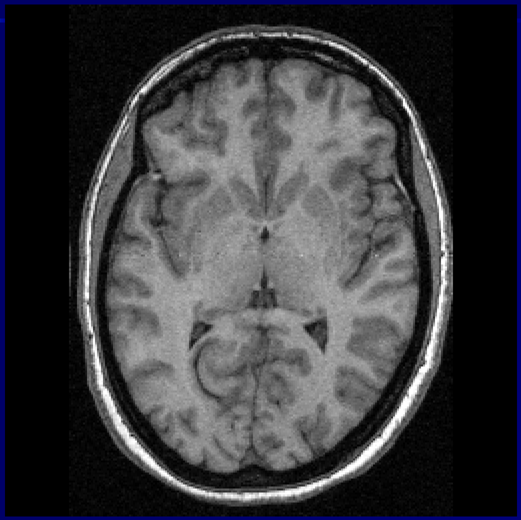


Removes most noise, with less effect on primary signal intensity.

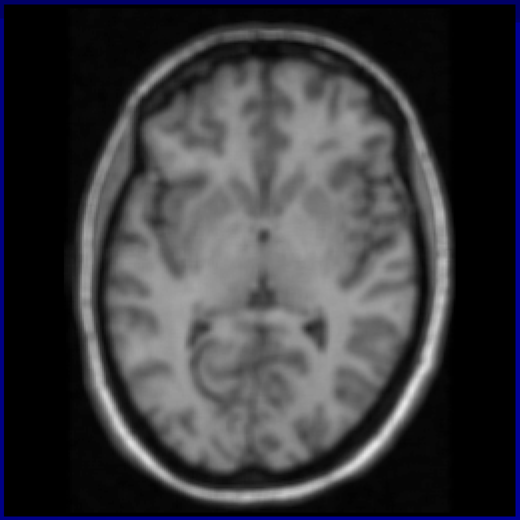


Noise removal – mean filters

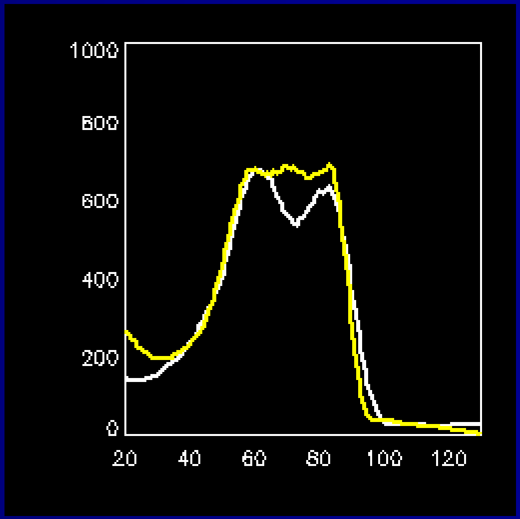
T1-weighted MRI slice



Boxcar mean, 5x5

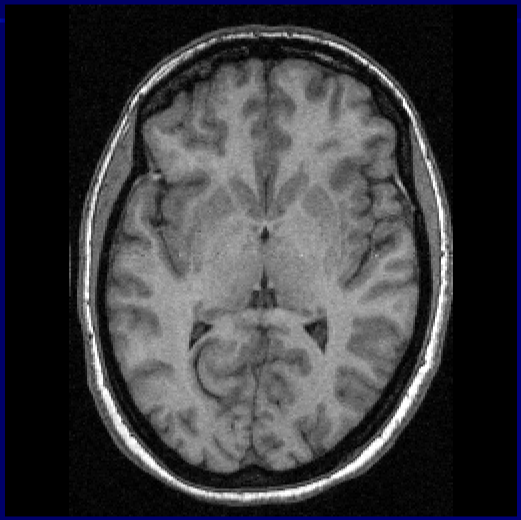


Distinct histogram peaks in original are poorly differentiated after smoothing.

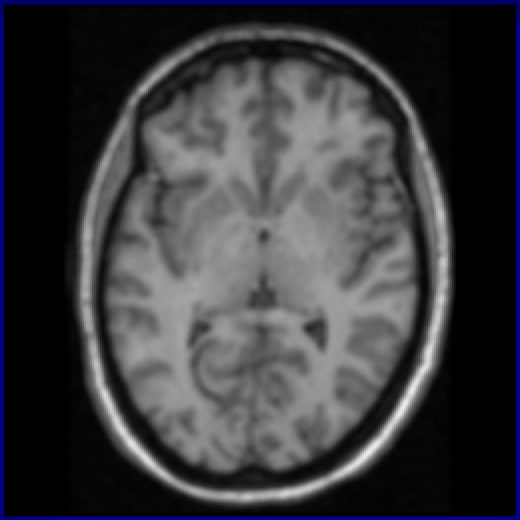


Noise removal – mean filters

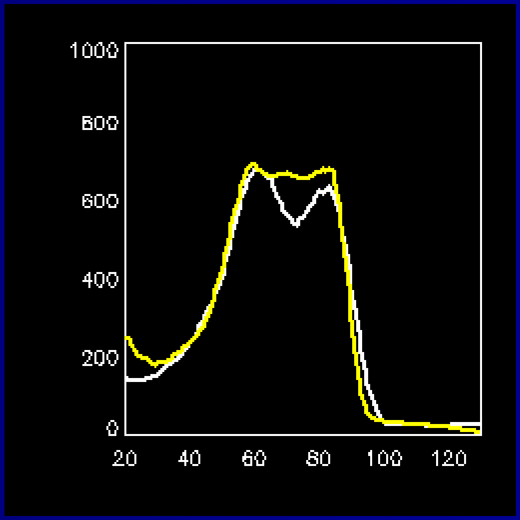
T1-weighted MRI slice



Gaussian mean, fwhm=5

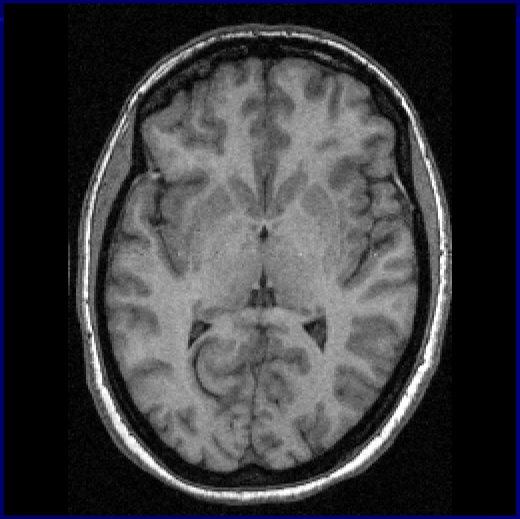


Note change in histogram peaks after smoothing.

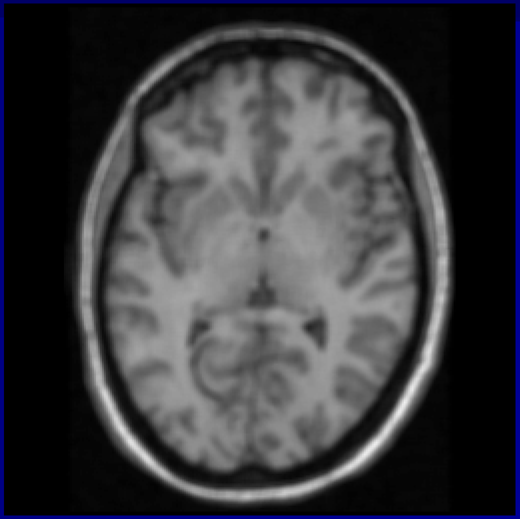


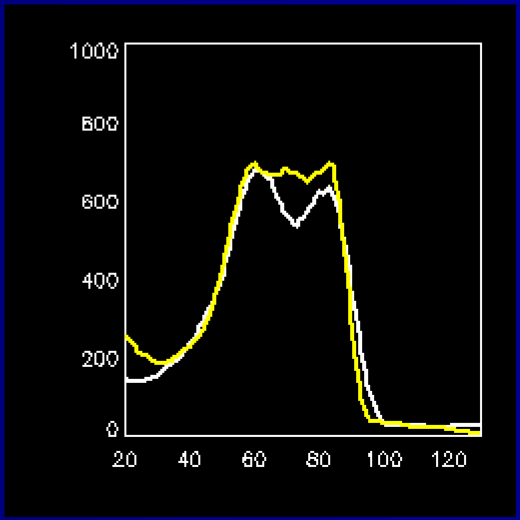
Noise removal – mean filters

T1-weighted MRI slice



Trimmed mean, 5x5





Noise removal – median filters

Median filters (non-linear)

- Gather N values using a kernel
- Sort by value
- Replace center pixel by median value

Removes features within the kernel that occupy less than N/2 pixels, also trims off edges of larger features. Can leave artifacts that reflect the kernel shape.

Noise removal – median filters

Median kernels; omni-directional

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

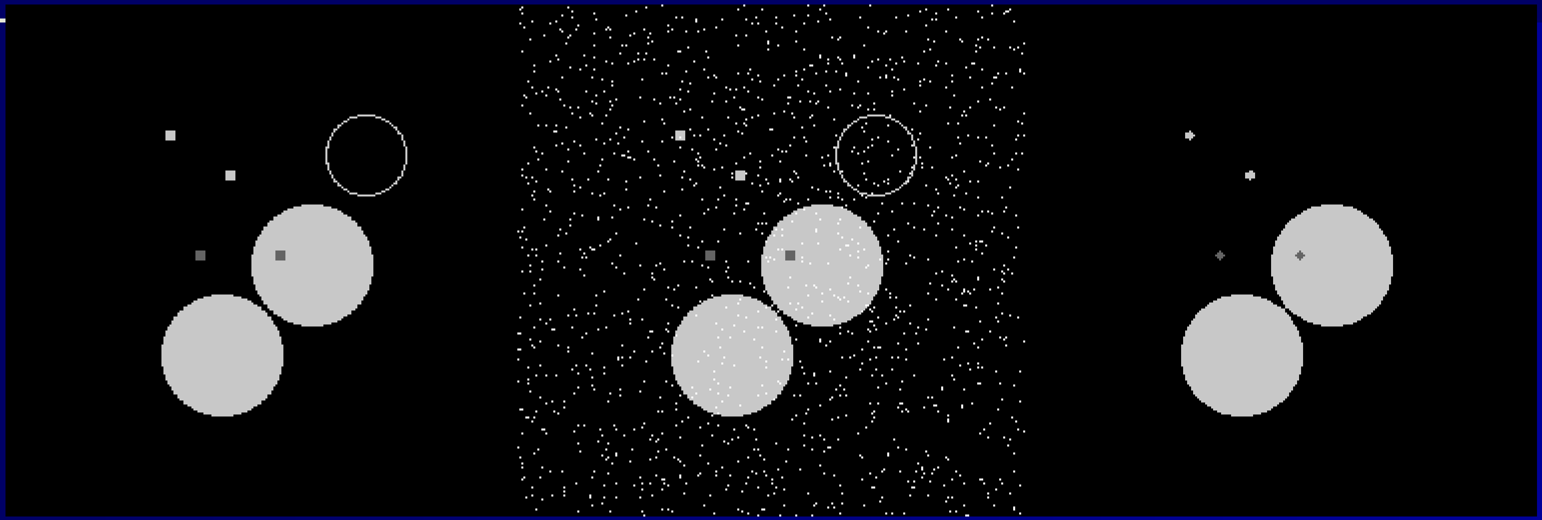
	1	1	1	
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
	1	1	1	

		1		
	1	1	1	
1	1	1	1	1
	1	1	1	
		1		

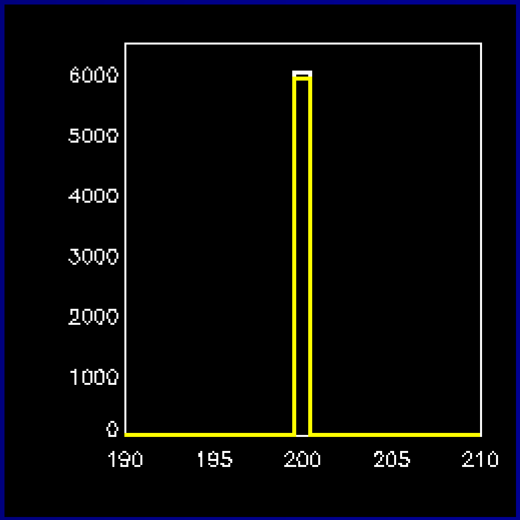
Center-weighting can be achieved by adding 2K copies of the central pixel prior to taking the median value.

Noise removal – median filters

Test object With additive noise Square median, 5x5



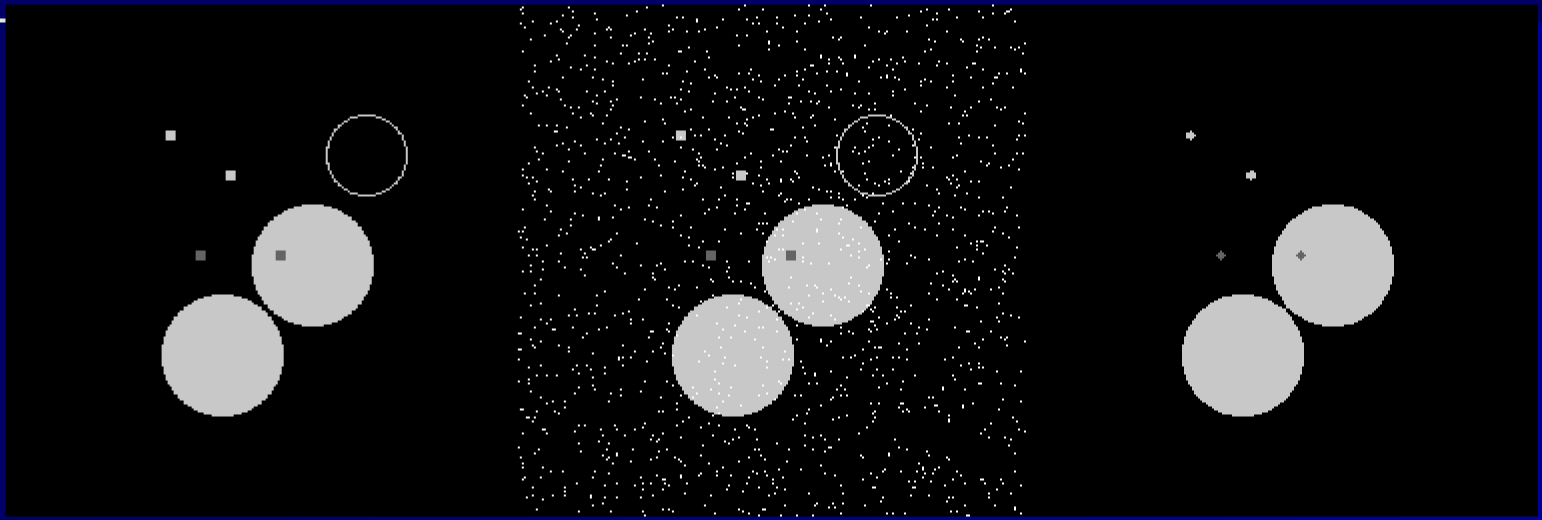
Noise is neatly suppressed – but also the open circle.



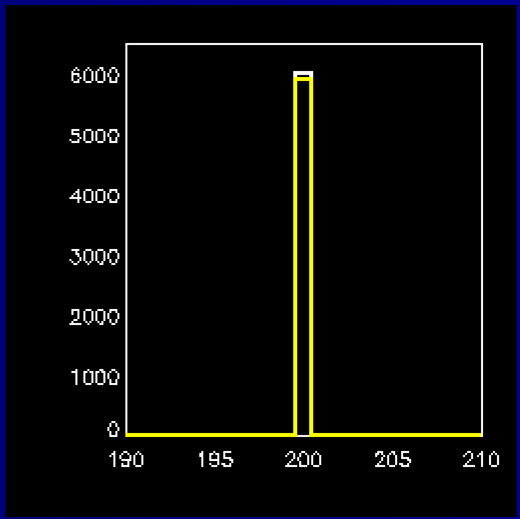
Noise removal – median filters

Test object

Center-weighted median, 5x5

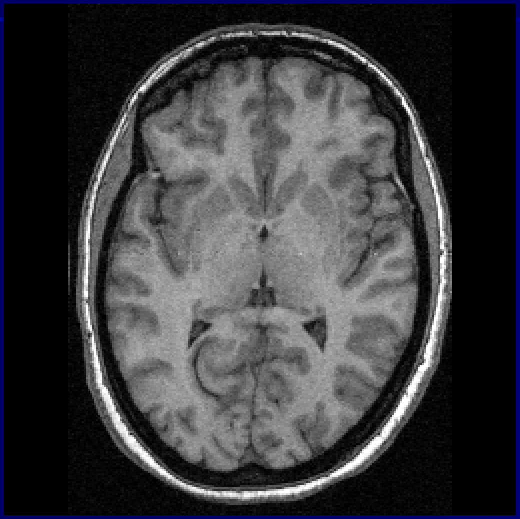


Center pixel got 3 votes instead of 1. No discernable difference from simple median.

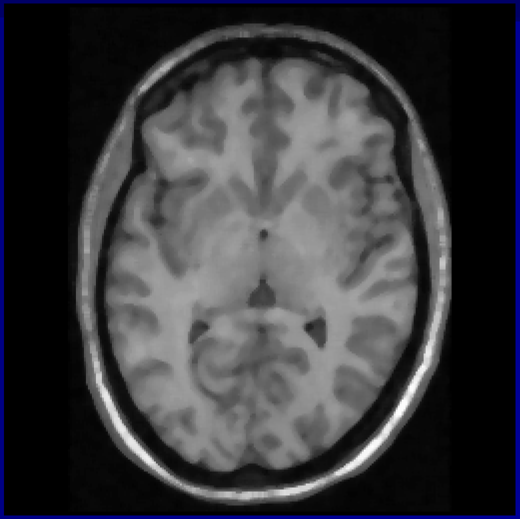


Noise removal – median filters

T1-weighted MRI slice

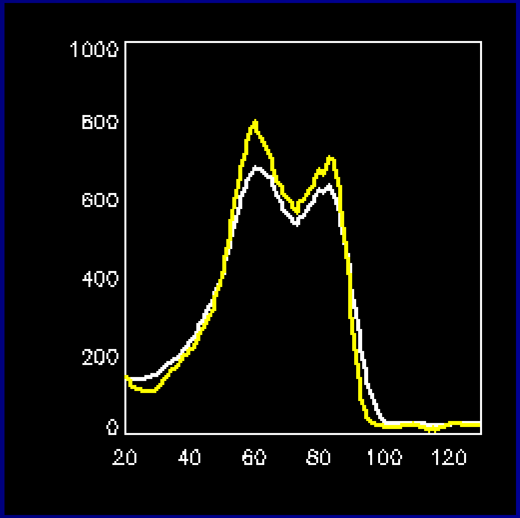


Square median, 5x5



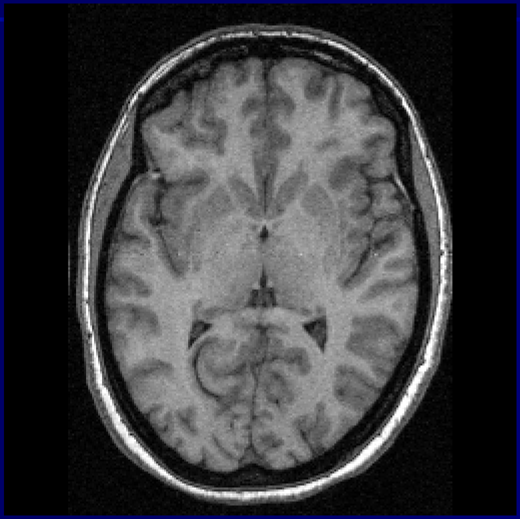
Note that bright rim, mid-hemisphere boundary have been suppressed.

Histogram peaks are accentuated, rather than smeared together.

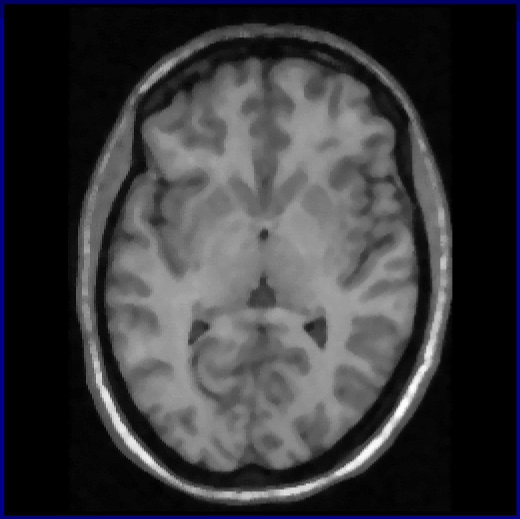


Noise removal – median filters

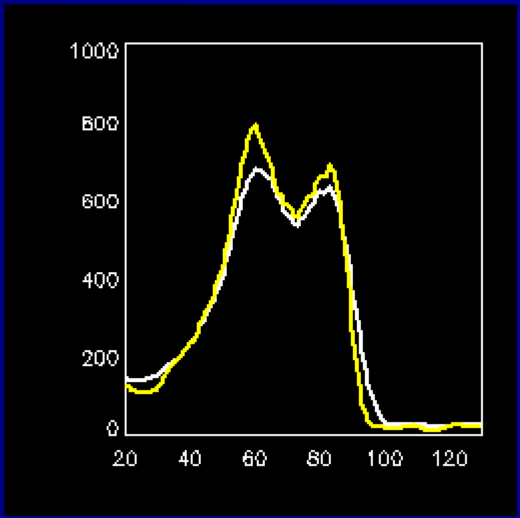
T1-weighted MRI slice



Center-weighted median, 5x5



Only subtly differs from ordinary median.



Noise removal – median filters

Median kernels; directional

		1		
		1		
		1		
		1		
		1		

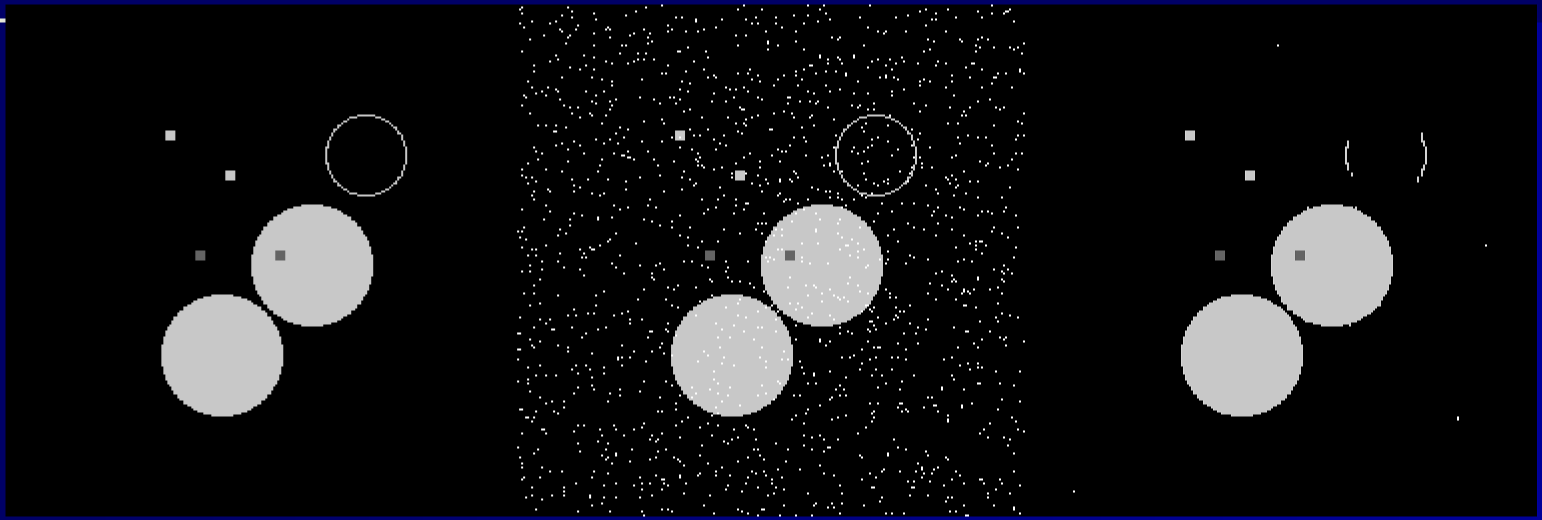
	1			
		1		
		1		
		1		
	1			

	1	1	1	
1				1

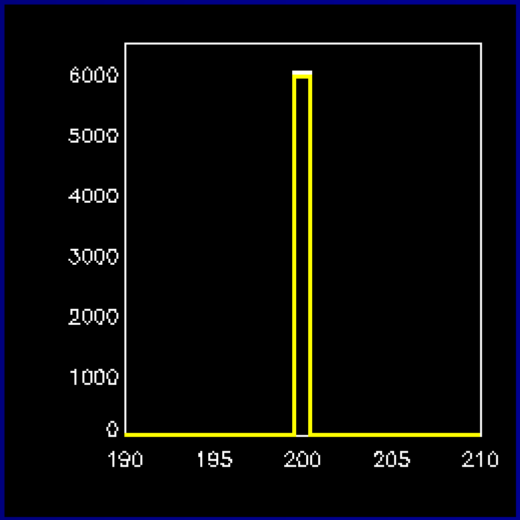
Noise removal – median filters

Test object

5-point vertical line median



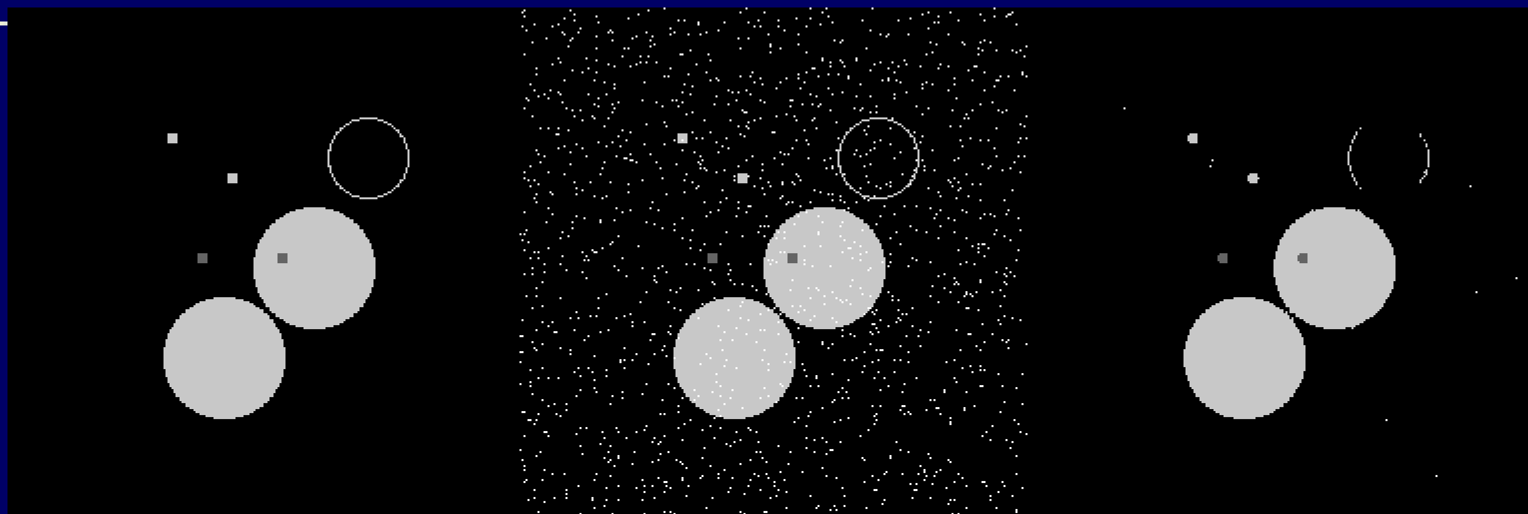
Retains a bit of the open circle; small signals keep their shape; a few noise specks remain.



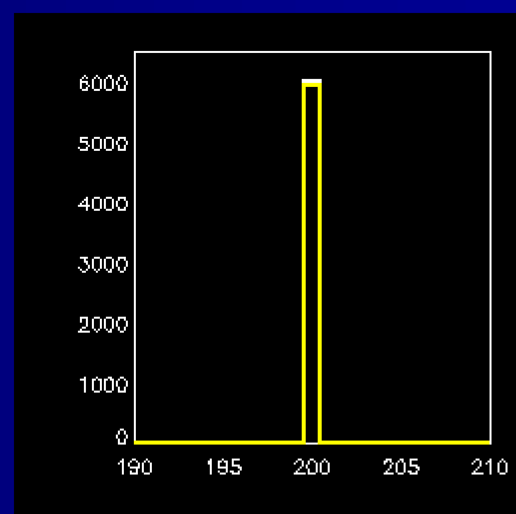
Noise removal – median filters

Test object

5-point curve median



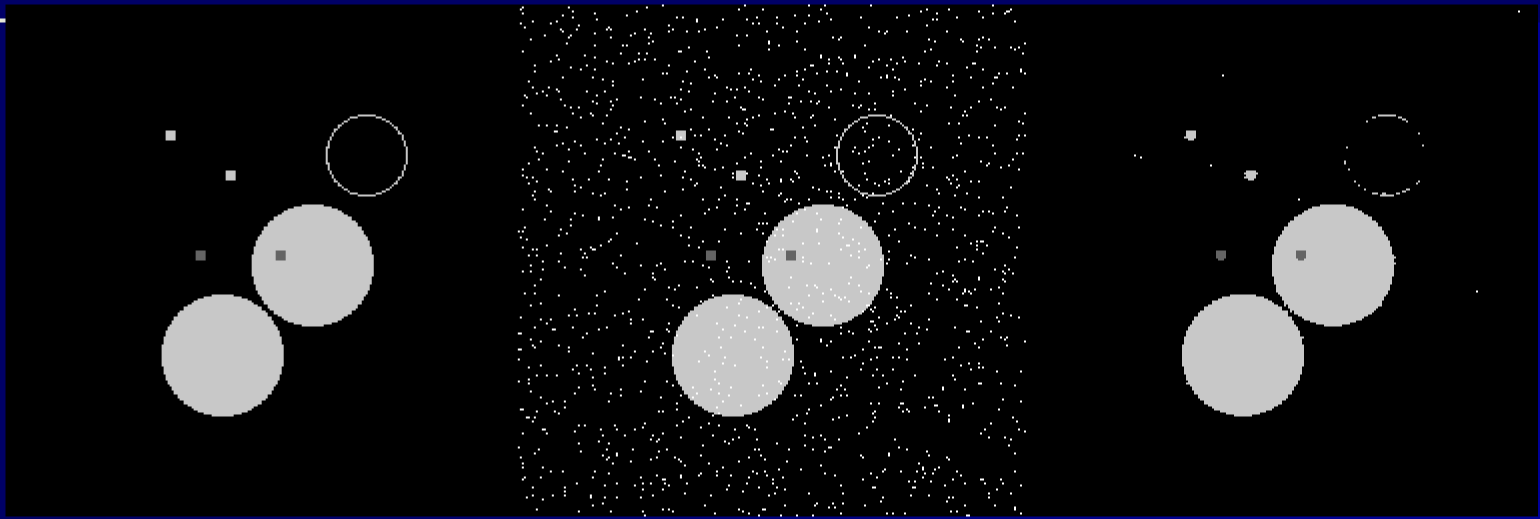
Retains more of the open circle; a few more noise specks remain.



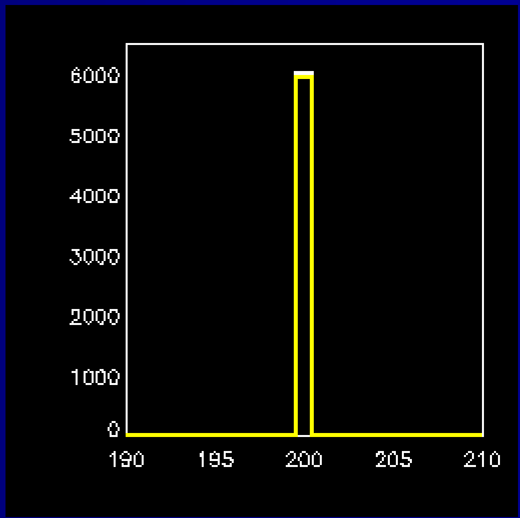
Noise removal – median filters

Test object

5-point rotated curve median

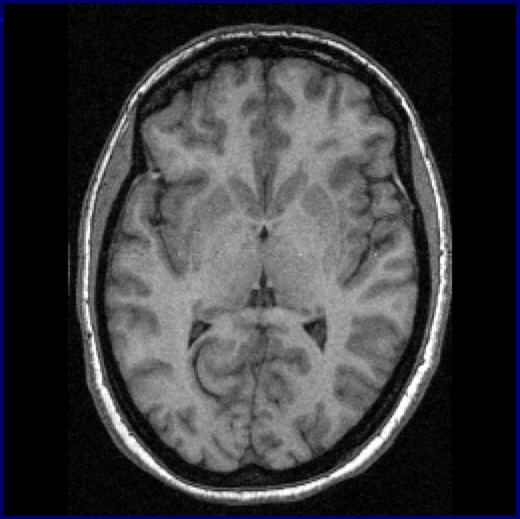


Retains different portions of the open circle; different noise specks survive.

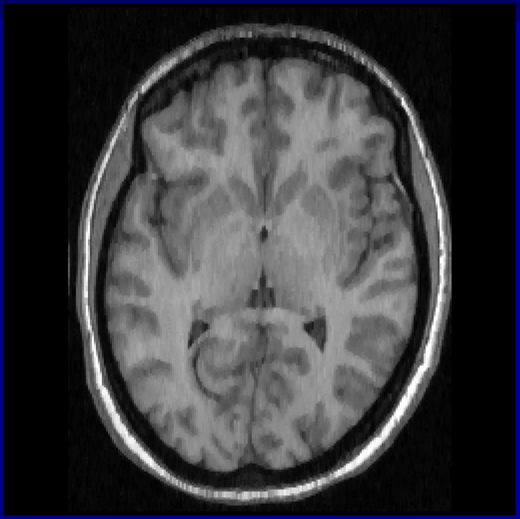


Noise removal – median filters

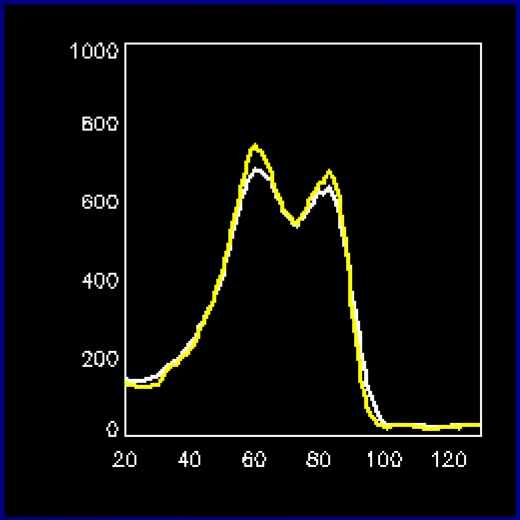
T1-weighted MRI slice



5-point vertical line median

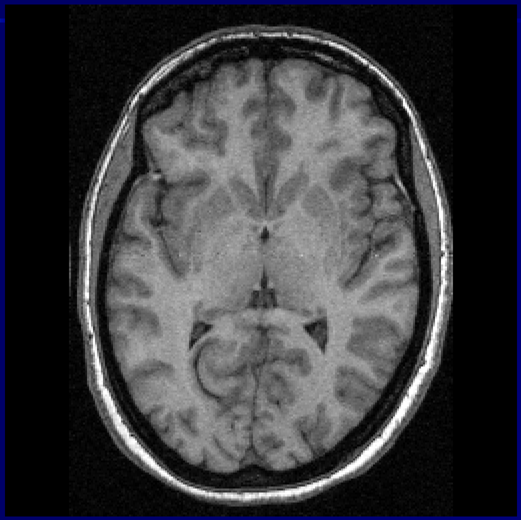


Rim is less degraded, histogram is similar to original.

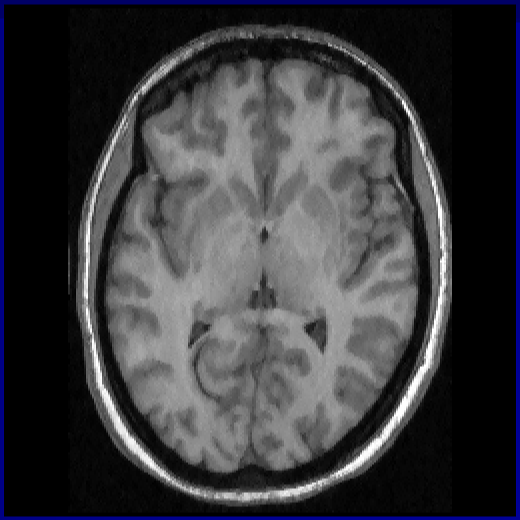


Noise removal – median filters

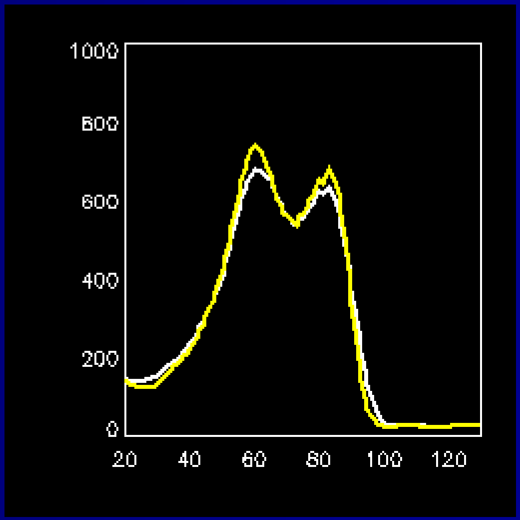
T1-weighted MRI slice



5-point curve median

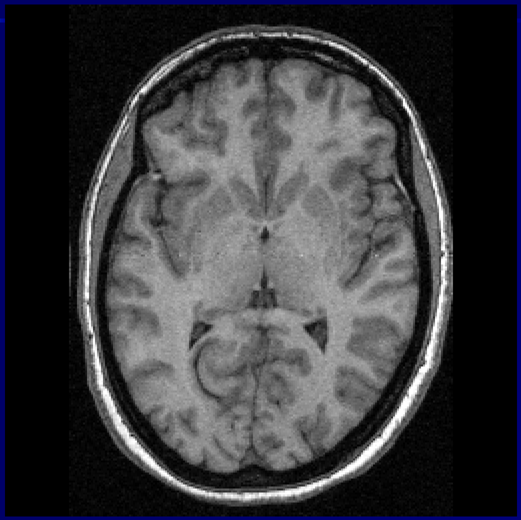


Similar to line kernel; left and right rims are well preserved.

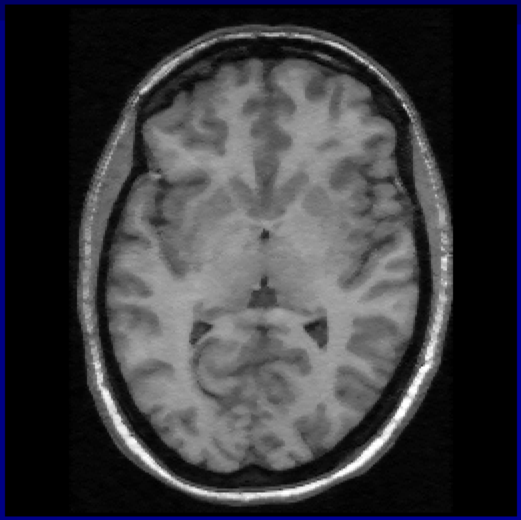


Noise removal – median filters

T1-weighted MRI slice

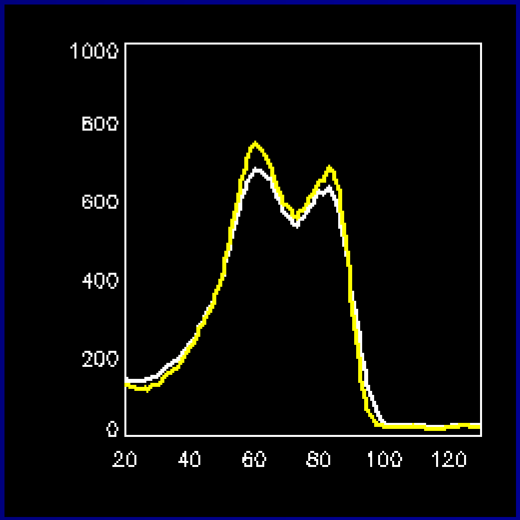


5-point rotated curve median



Now it's the fore and aft rim that is well-preserved.

Note that mid-hemisphere boundary is less distinct.



Noise removal – Wiener filter

Wiener filter attempts to minimize mean-square-error

$$g = f + n \qquad \hat{f} = h * g \qquad \hat{F} = HG$$

$$H = \frac{S_{ff}}{S_{ff} + S_{nn}}$$

where S_{ff} is the Fourier transform of the autocorrelation function of the image

and, if we have stationary white noise, $S_{nn} = s_n^2$

(known or estimated from a section of the image)

Noise removal - anisotropic smoothing

2D iterative method for
"diffusion-weighted" smoothing

	N	
E	T	W
	S	

T' = T + p*(Cn*(N-T) + Cs*(S-T)) +
p*(Ce*(E-T) + Cw*(W-T))

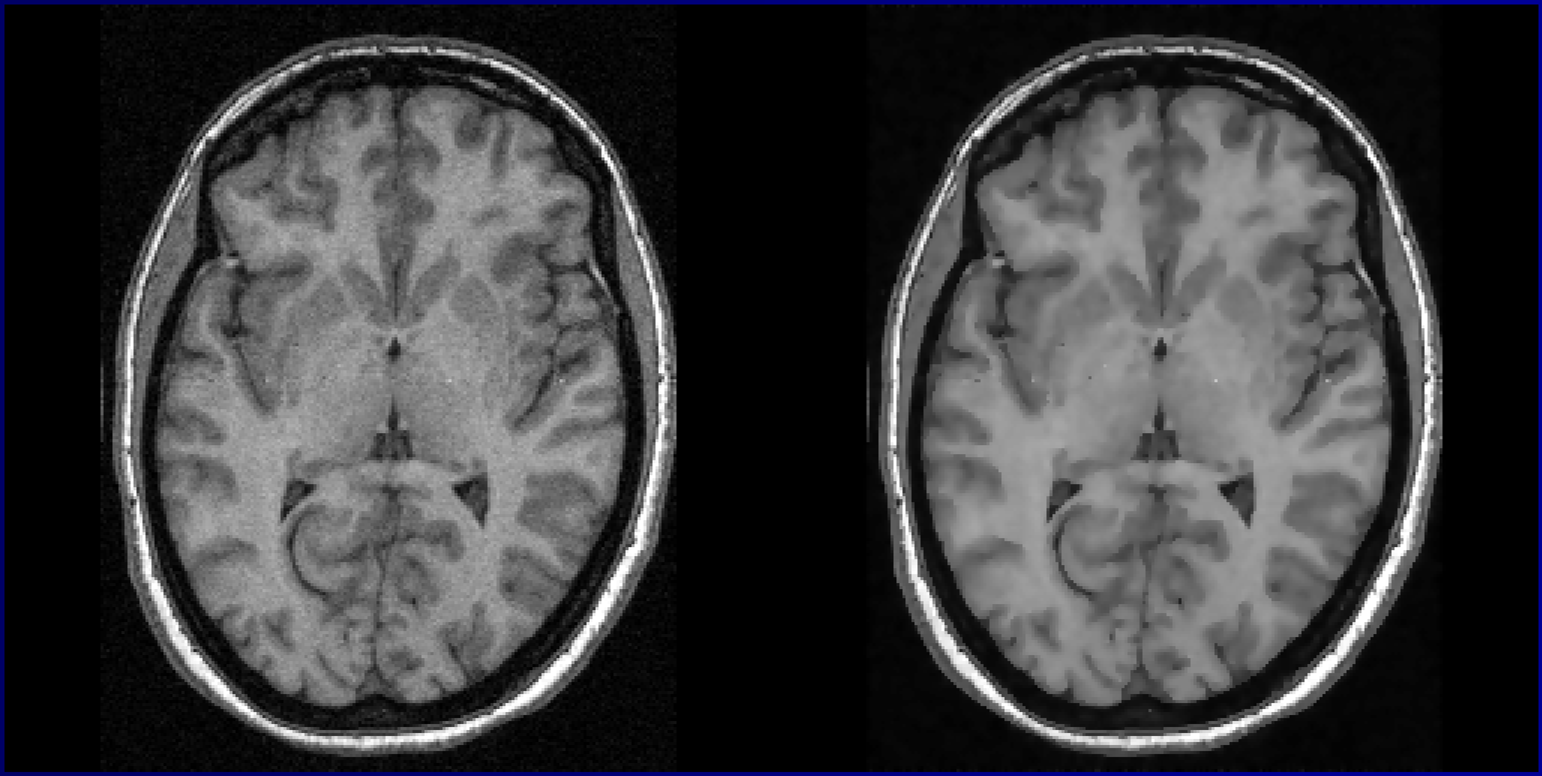
Cn = exp (-1 * (|N-T|/K)^2)

When gradient is large, conduction coefficient is negligible, so that neighbor can not affect the value of T. Extend to 3D by including Up and Down terms.

Noise removal - anisotropic smoothing

Raw

Anisotropic smoothing, K=10, iterations = 6



How to filter to …

- Increase contrast
- Remove noise
- Emphasize edges
- Detect edges
- Modify shapes

Emphasize edges – spatial domain

Edge-boost methods use directional or omnidirectional kernels to identify edges; these can be recombined as

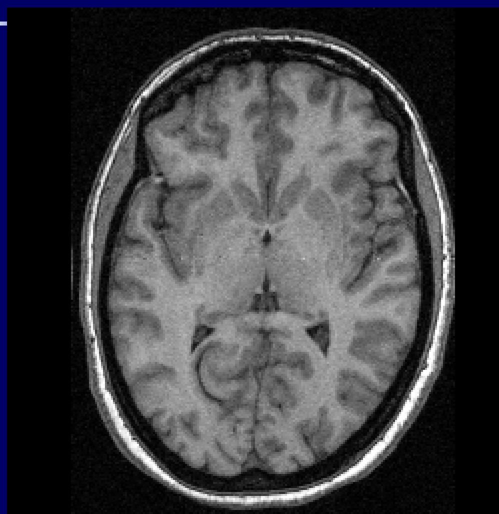
$$g = \text{image} + a(\text{edges})$$

Unsharp masking uses a convolution kernel to compute a very blurred version of an image; the enhanced image is defined as

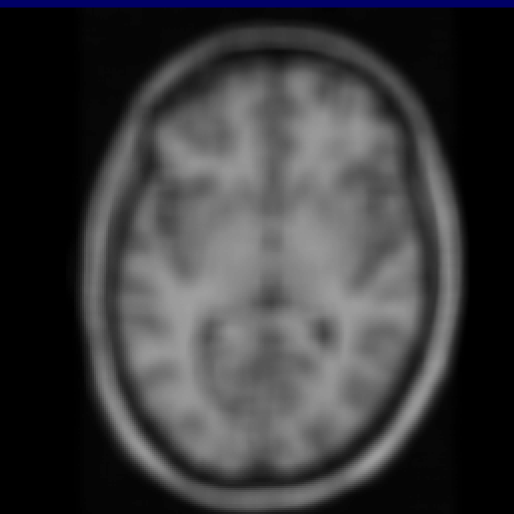
$$g = \text{unsharp} + b(\text{image} - \text{unsharp})$$

Emphasize edges – spatial domain

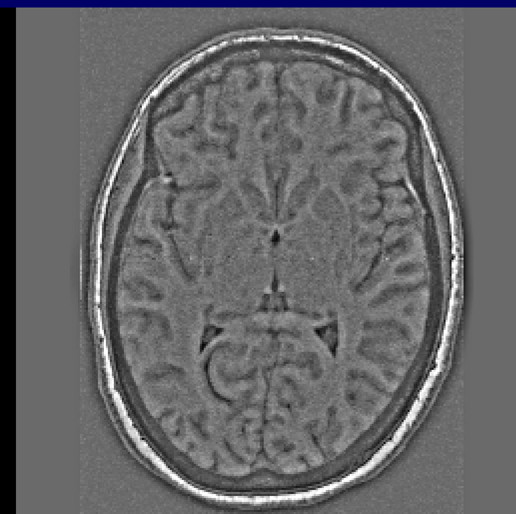
Original

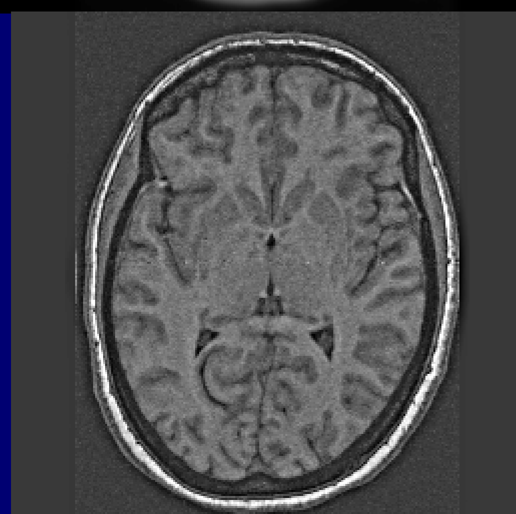


Unsharp



Difference





Emphasize edges – frequency domain

Compute Fourier transform of image

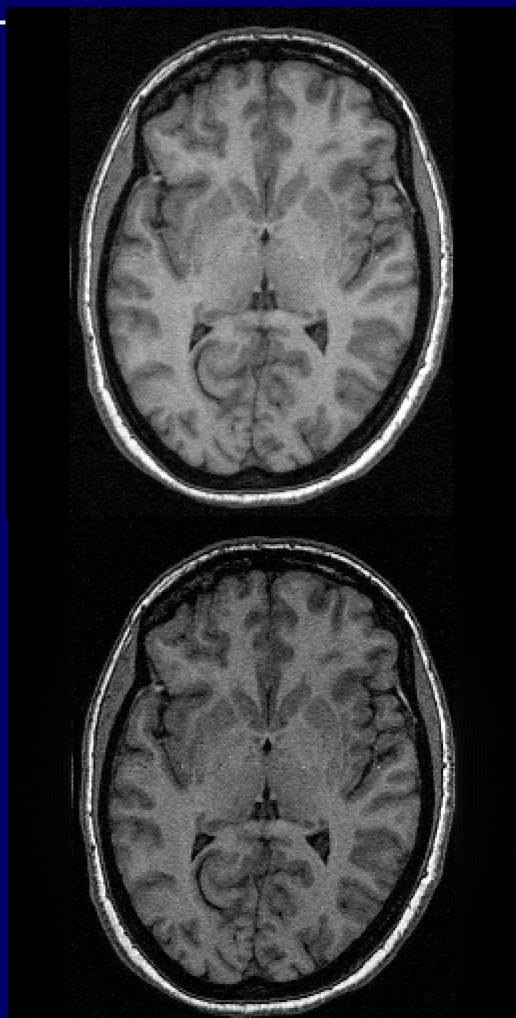
Apply a weighting function to boost high frequency components

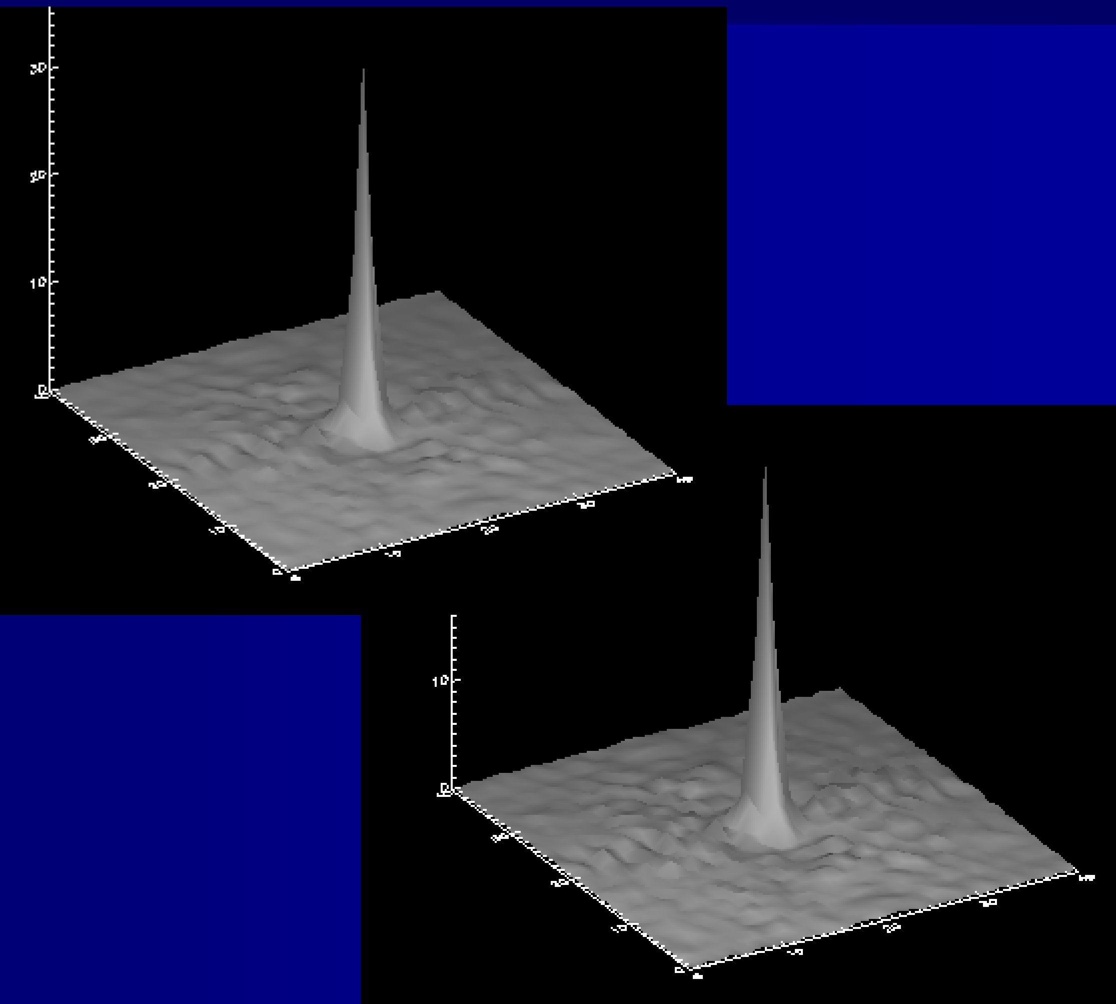
Compute inverse transform

Emphasize edges – frequency domain

Original, boosted

FFT (original, modified)





How to filter to …

- Increase contrast
- Remove noise
- Emphasize edges
- Detect edges
- Modify shapes

Edge detection

- Horizontal, vertical, omni-directional kernels
- Sobel filter
- Roberts' cross
- Difference of Gaussians
- Marr-Hildreth filter

Edge detection

Convolution kernels that take derivatives

Gx = horizontal derivative:

-1	0	1
-2	0	2
-1	0	1

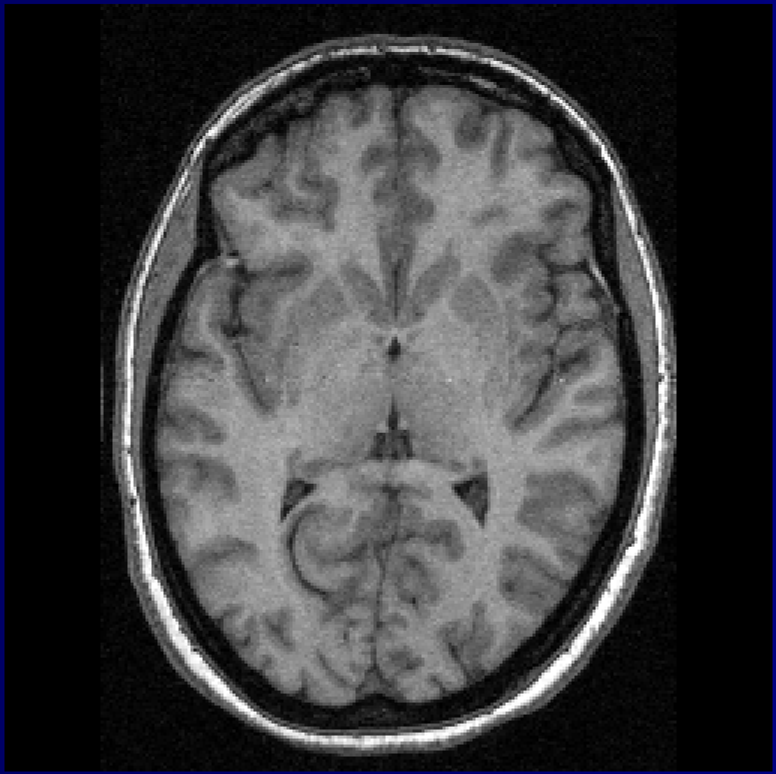
Gy = vertical derivative:

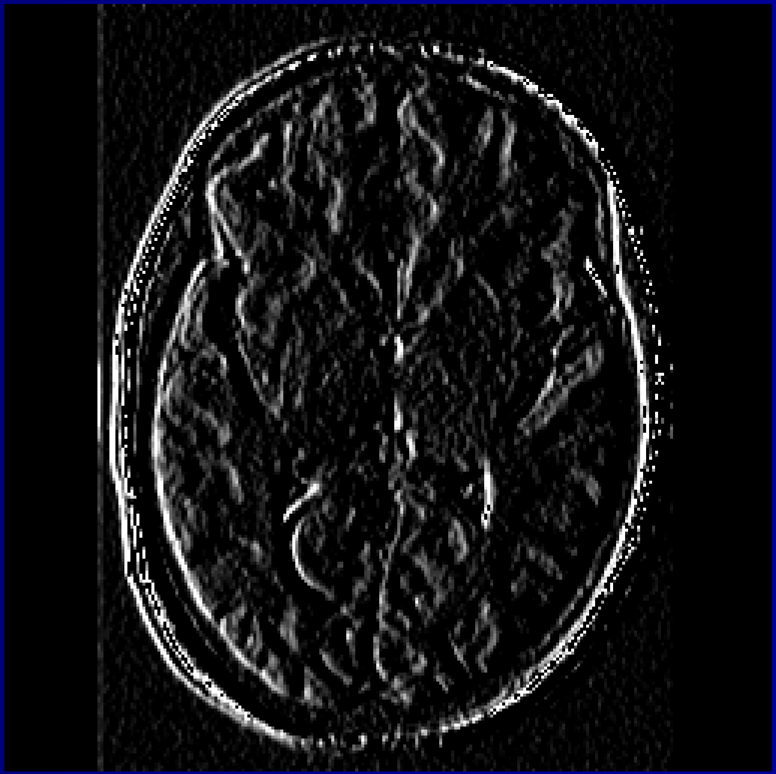
-1	-2	-1
0	0	0
1	2	1

Laplacian = second derivative

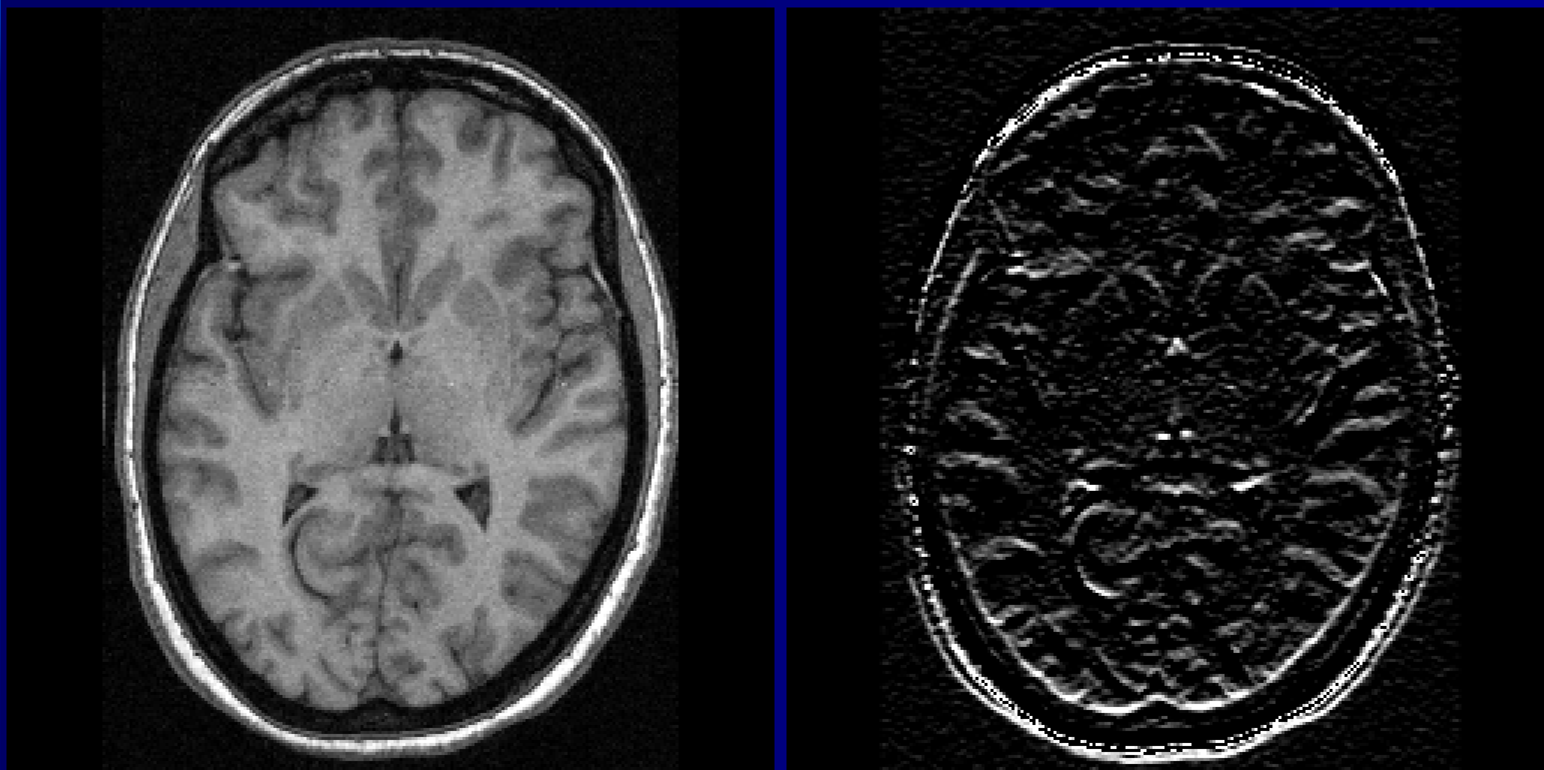
-1	-1	-1
-1	8	-1
-1	-1	-1

Edge detection – horizontal derivative

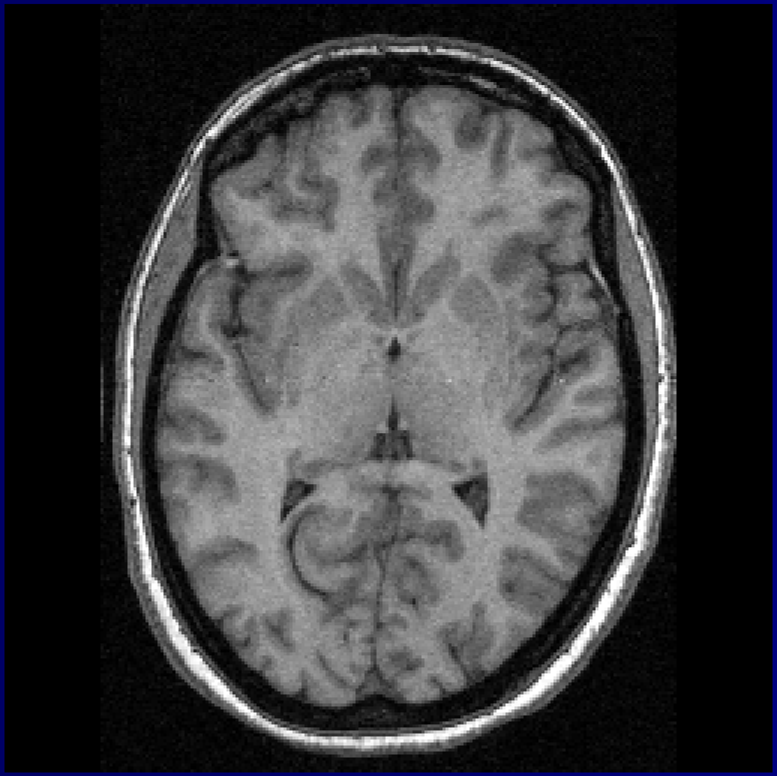


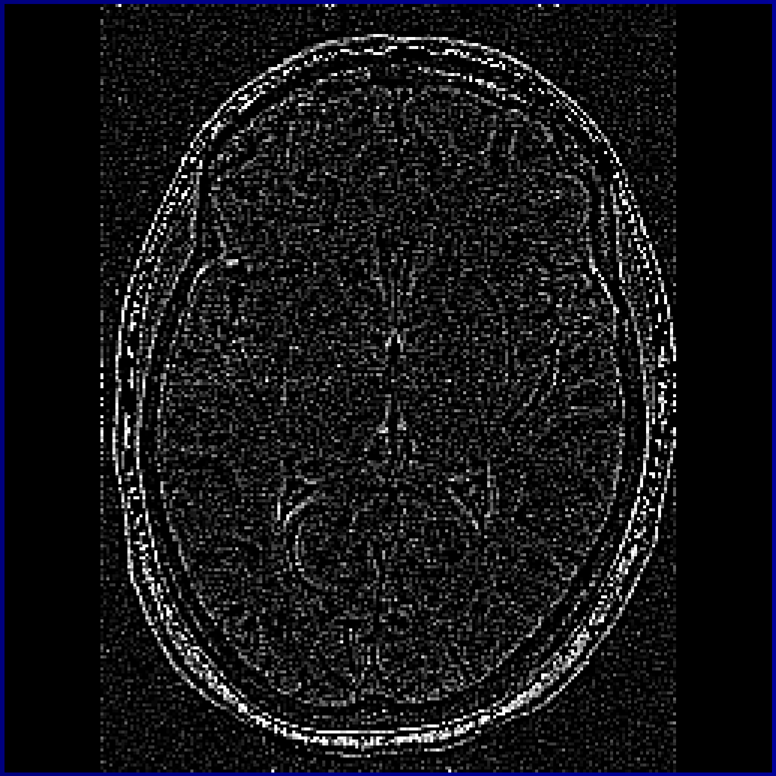


Edge detection – vertical derivative



Edge detection – Laplacian





Edge detection

Nonlinear methods often combine directional derivatives.

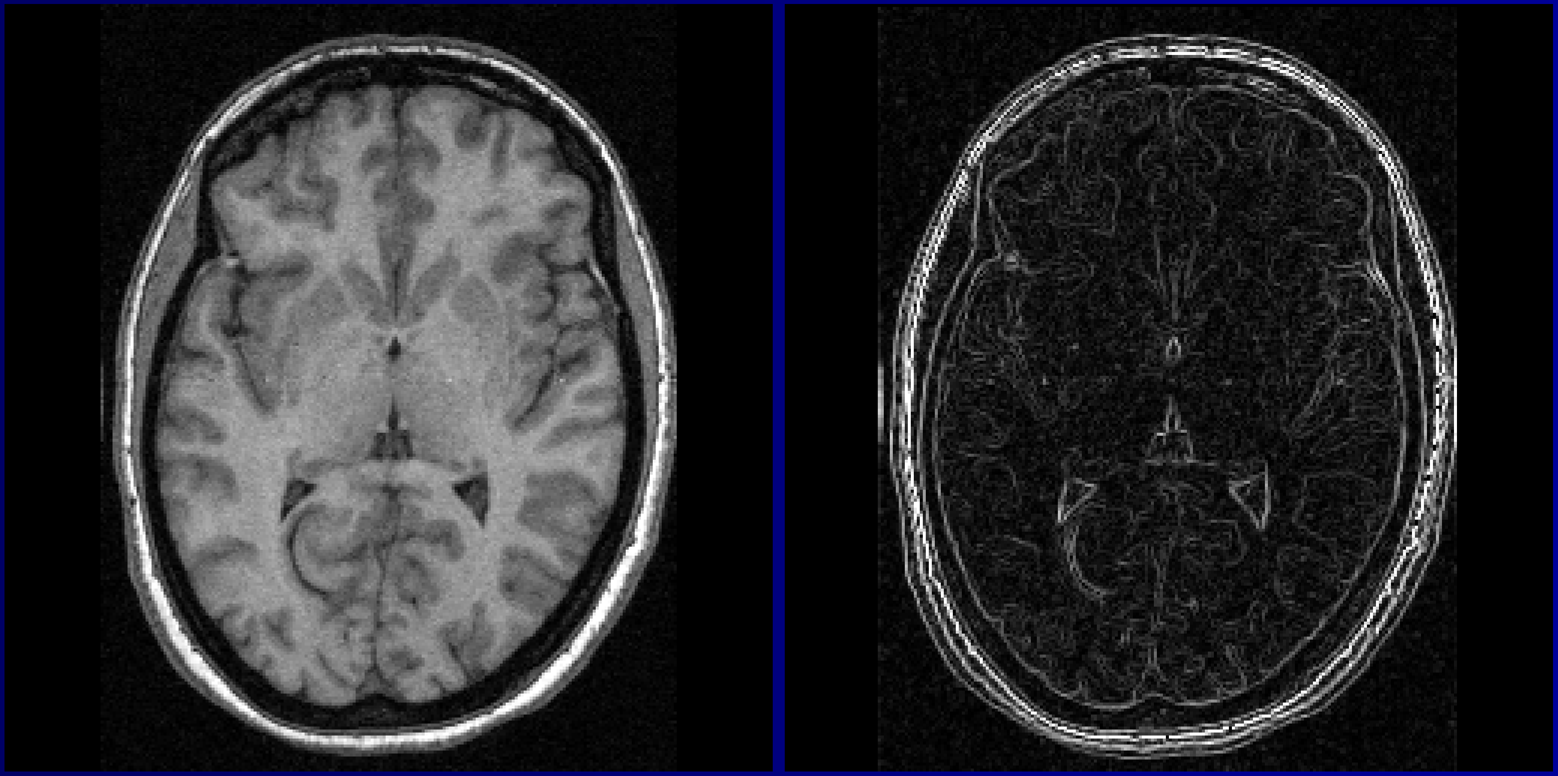
Pixel labels (2D) :

F	A3
A5	A4

Roberts' cross:

$$\text{result} = \text{SQRT}(\ [F - A4]^2 + [A3-A5]^2\ )$$

Edge detection – Roberts'



Edge detection

Pixel labels (2D) :

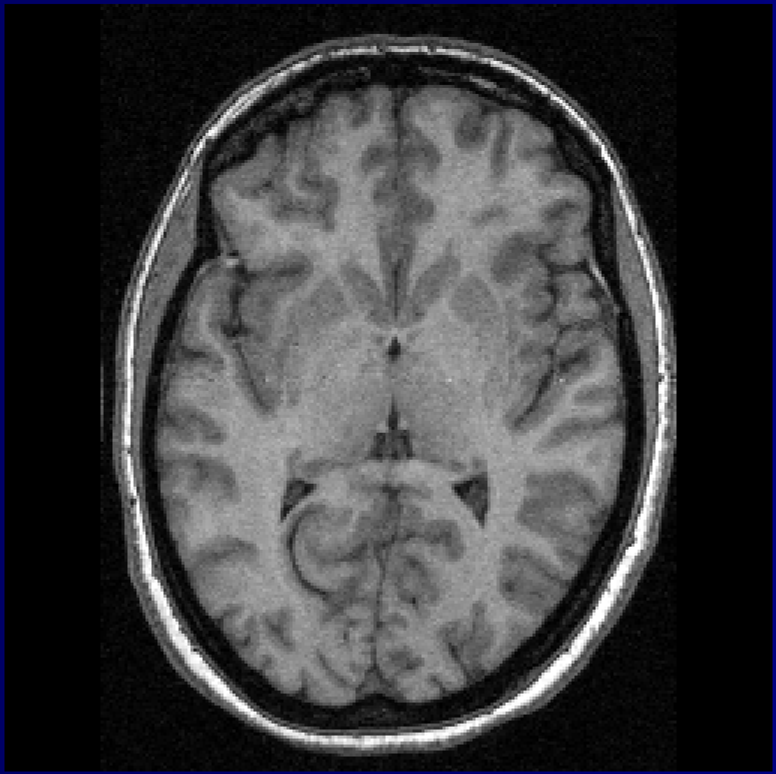
A0	A1	A2
A7	F	A3
A6	A5	A4

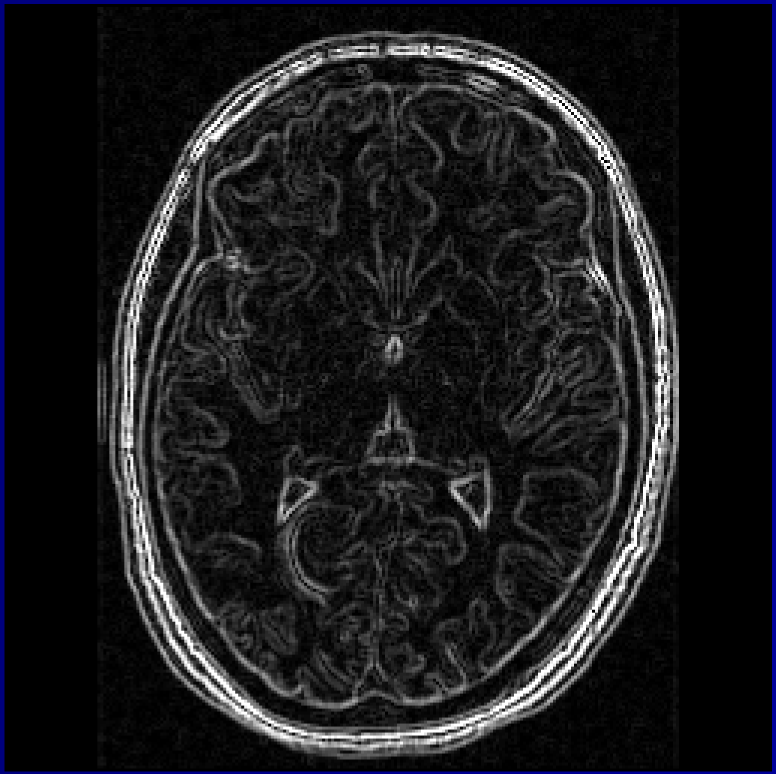
Sobel operator: result = SQRT(X*X + Y*Y)   where

X = (A2+2A3+A4) - (A0+2A7+A6)

Y = (A0+2A1+A2) - (A6+2A5+A4)

Edge detection – Sobel



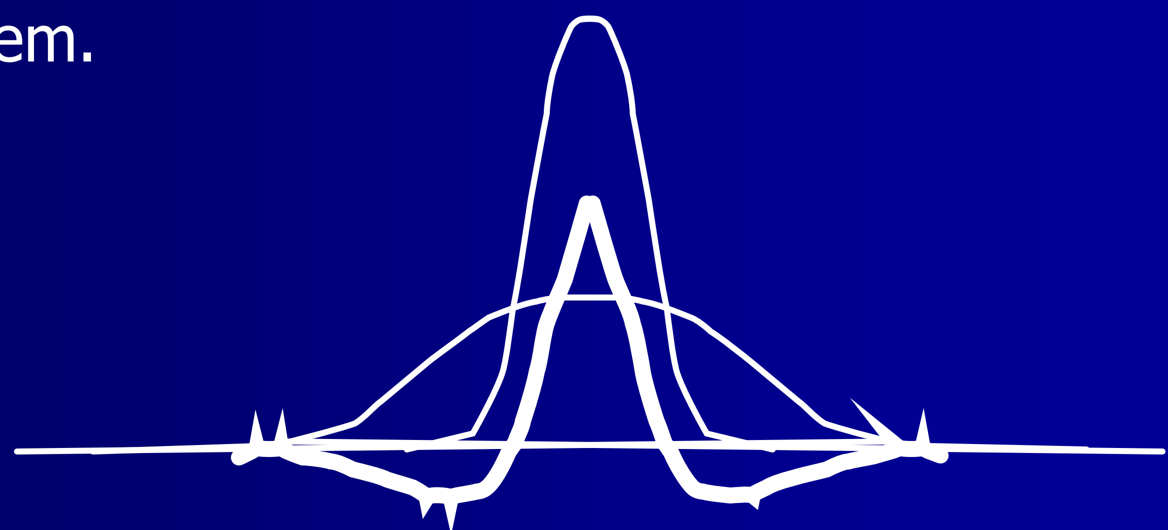


Edge detection

Difference of Gaussians ("DOG")

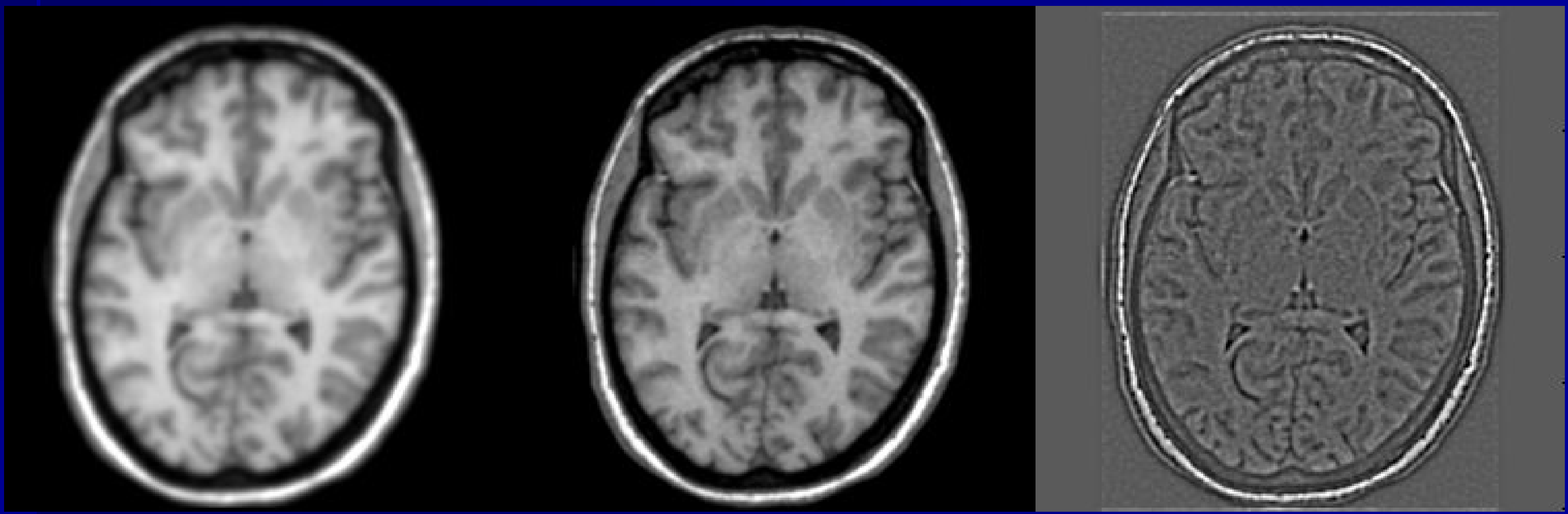
$$\text{result} = \text{Gauss}(s1)_{**}\text{image} - \text{Gauss}(s2)_{**}\text{image}$$

This is believed to be similar to processing in the human visual system.



Edge detection - DOG

Gaussian 1 | Gaussian 2 | Difference



Edge detection

Marr-Hildreth operator -- filter is a Laplacian of a Gaussian

$$\text{2D: } C(x,y) = \nabla\ (I(x,y) ** G(x,y,r))$$

$$\text{3D: } C(x,y,z) = \nabla\ (I(x,y,z) ** G(x,y,z,s))$$

$$= I(x,y,z) ** \nabla\ (G(x,y,z,s))$$

Result is very similar to DOG filtering

Edge detection – zero crossings

2D pixel labeling:

A0	A1	A2
A7	F	A3
A6	A5	A4

If A0 & F are different signs, Grad1 = |A0 - F|, else Grad1 = 0.0

If A1 & F are different signs, Grad2 = |A1 - F|, else Grad2 = 0.0

.

.

.

Result = Maximum of Grad1 ... Grad8

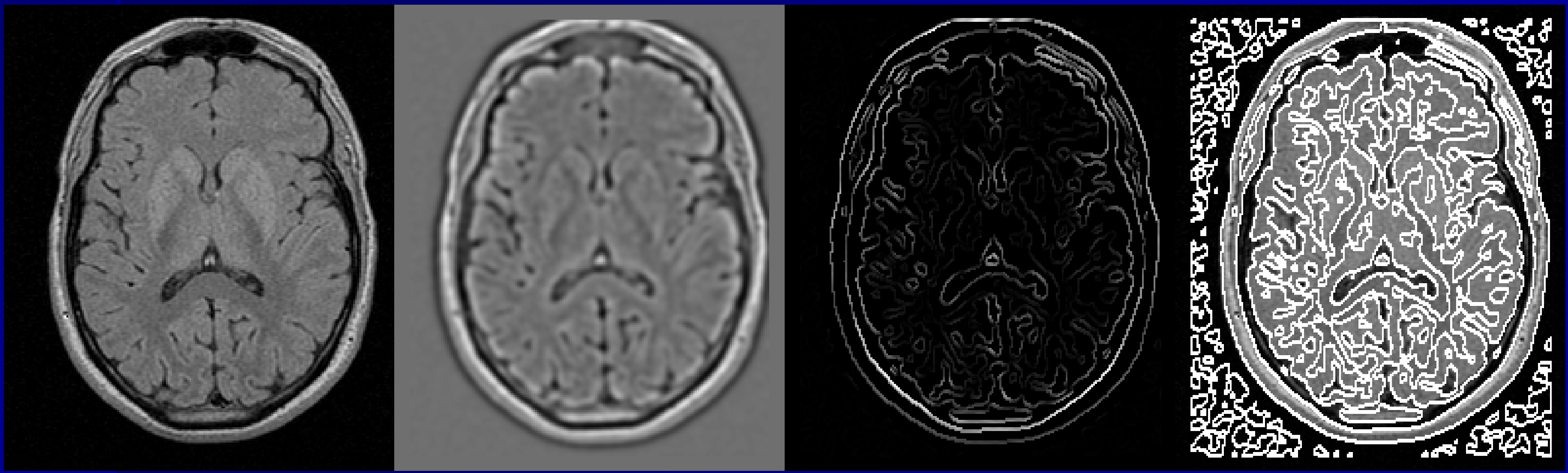
Edge detection

Original

Filtered with 2D Marr-Hildreth operator

Zero-crossings

Zero-crossings overlaid on original data



Edge detection

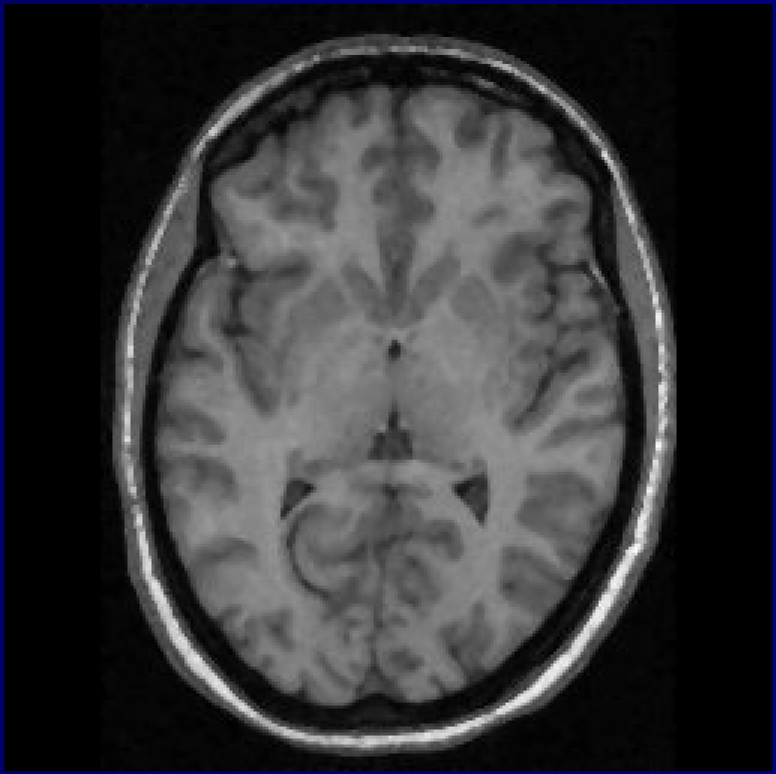
Frei and Chen filter:

This sums the result of 9 convolution filters, then takes the COS(SQRT(sum)).

The first kernel is a boxcar average; the others are various derivative kernels

Ref. J.C. Russ, *The Image Processing Handbook,* CRC Press, 1995

Edge detection – Frei and Chen



How to filter to …

- Increase contrast
- Remove noise
- Emphasize edges
- Detect edges
- Modify shapes

Shape modification tools

Morphological (shape-based) operations

- Erosion
- Dilation
- Opening
- Closing

Have predictable effects on shape.

Morphological operators

Erosion ("thin" or "remove outer layer") Ⓧ

-- Gather N values using a kernel

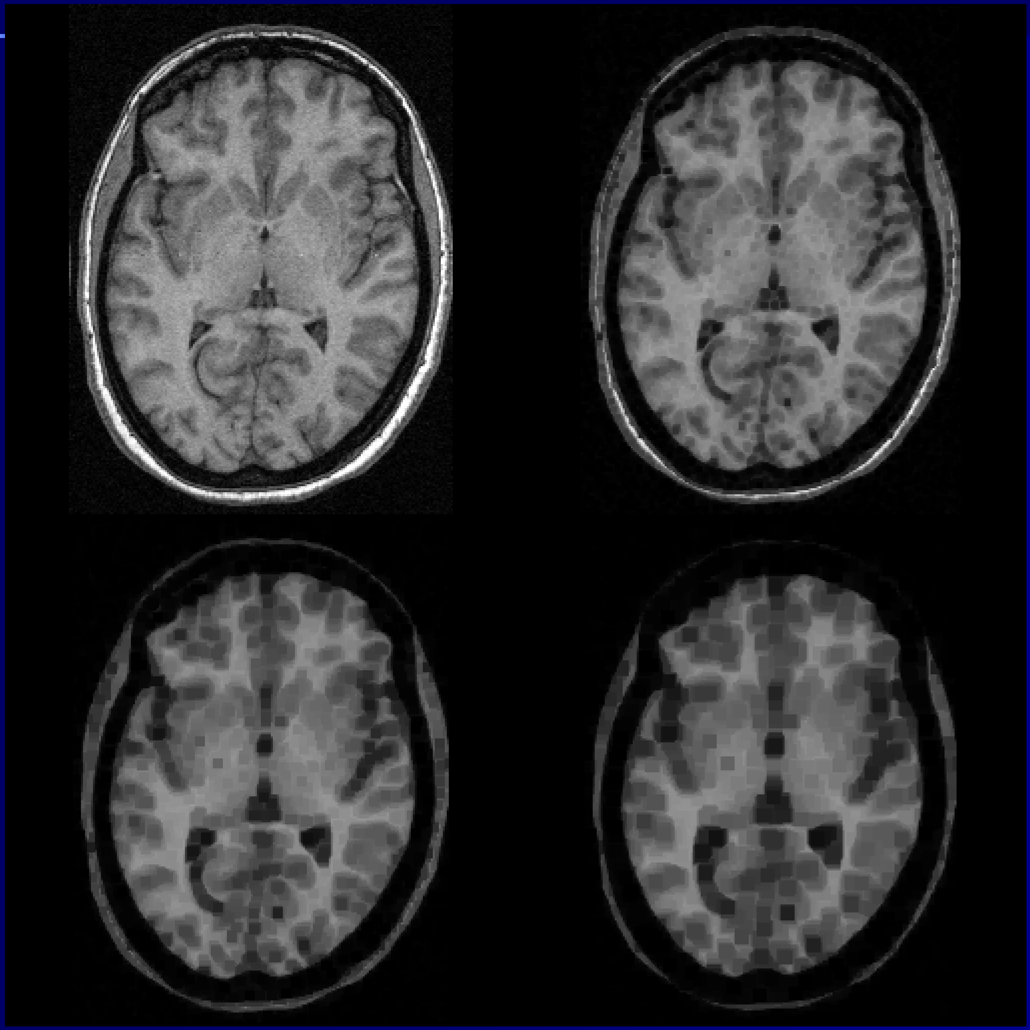
-- Sort by value

-- Replace center pixel by minimum value

Islands smaller than the kernel are removed.

Small pennisulas are broken.

Morphological operators - erosion



raw,
3x3 erosion

eroded 2 times,
eroded 3 times

Morphological operators

Dilation ("grow" or "fill holes") $\oplus$

-- Gather N values using a kernel

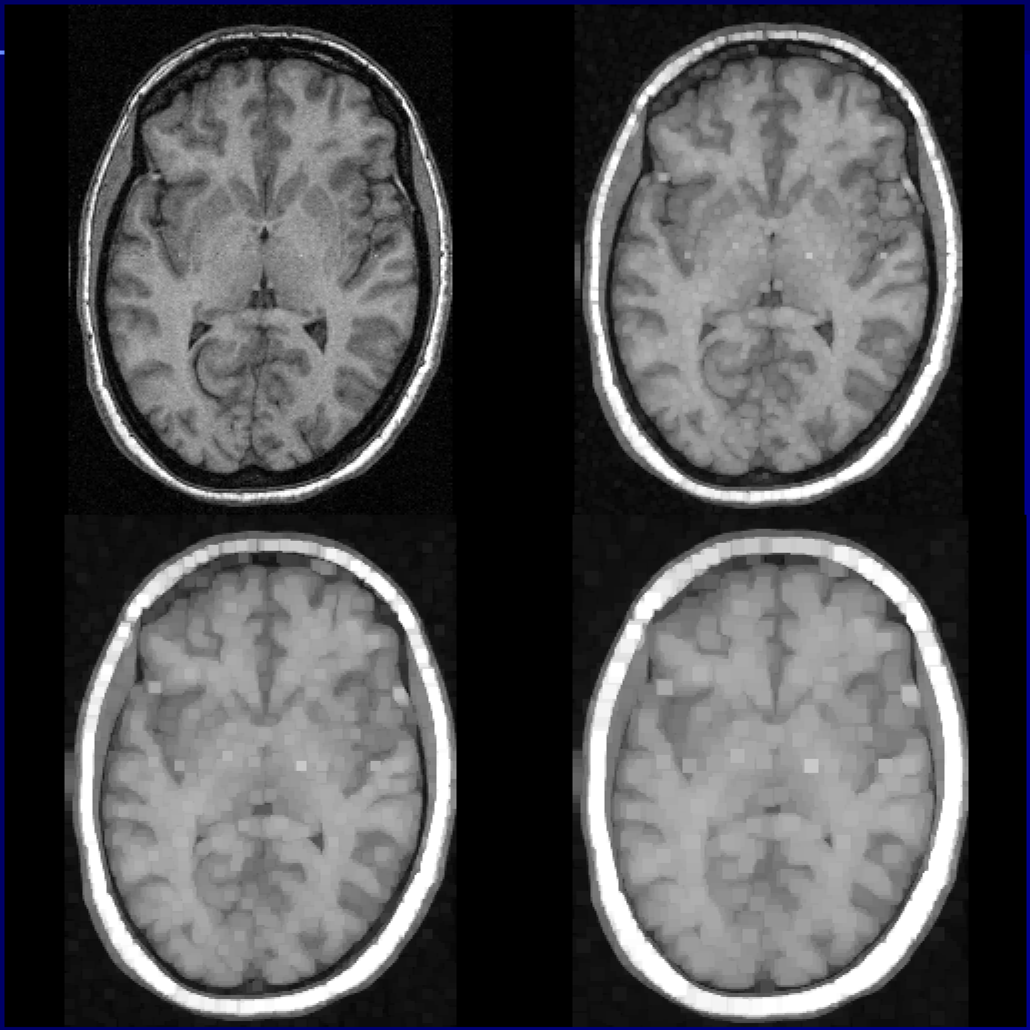
-- Sort by value

-- Replace center pixel by maximum value

Holes smaller than the kernel are filled.

Neighboring islands may be connected.

Morphological operators - dilation



raw,
3x3 dilation

dilated 2 times,
dilated 3 times

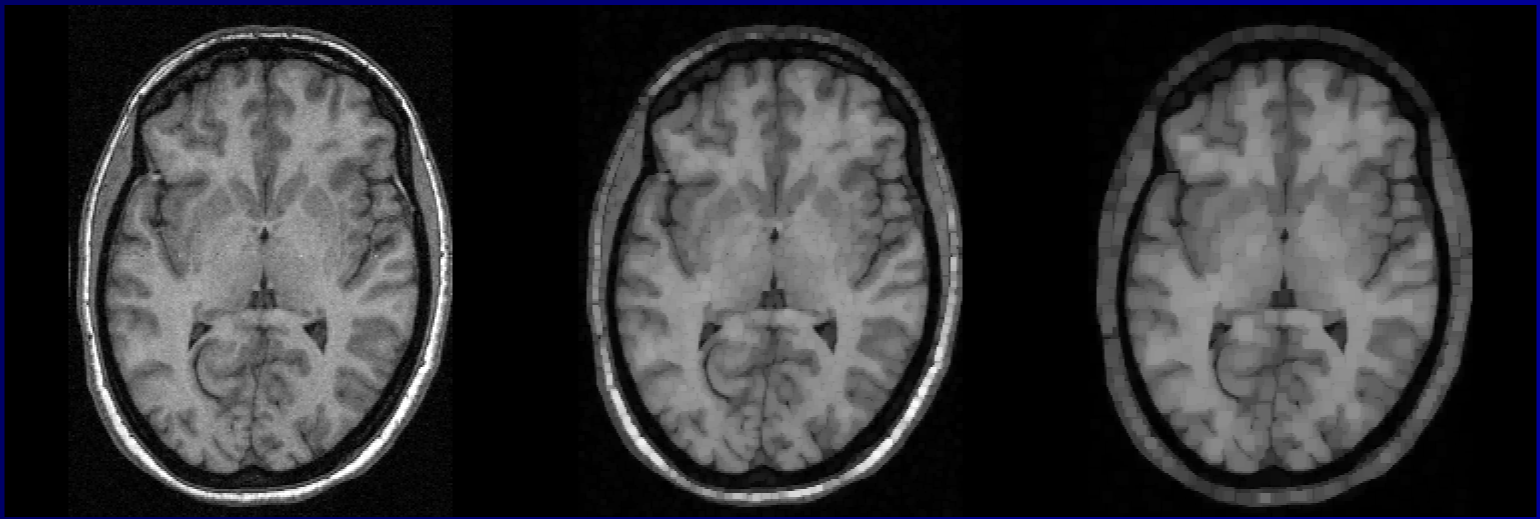
Morphological operators

Opening = dilation followed by erosion

Smoothes a contour
Breaks narrow isthmuses
Eliminates small islands
Eliminates sharp capes

Morphological operators - opening

raw | open (3x3) | open(5x5)



Morphological operators

Closing = erosion followed by dilation

Smoothes a contour
Fuses narrow breaks
Fills small holes
Fills small gaps on a contour

Morphological operators - closing

raw close (3x3) close(5x5)